PERSONALNI
računari
76
NOVEMBAR
1991.
CENA 120 D

SOFTVER

AUTOCAD AME
AUTOCAD AEC
CLIPPER++
CODE BASE
LOTUS 2.3

JEZICI

LAKI UVOD U ADU

OPERATIVNI SISTEMI

UNIX ZA POČETNIKE



HARDVER

TEST VGA KARTICA
TEST TVRDIH DISKOVA
EPSON EL3/33

YU ISSN 0352-7271
9 770352 727009

IBIS SYS
ekonomski biro

ima bolju ideju . . .

POZIVAMO

— sve korisnike PC-računara nezadovoljne servisnom podrškom svojih dobavljača,

— sve distributere i dealere PC-računara, koji imaju teškoća u organizovanju sopstvenog servisa u garantnom i vangarantnom roku,

NA DOGOVOR

Tehnička laboratorija

IBIS SYS
SOFTWARE HARDWARE INŽENJERING

servisira
dograđuje
zamenjuje **RAČUNARE**
povezuje

Telefon tehničke laboratorije: 011/609-650, 605-890

PERSONALNI
računari 76

časopis za informatiku i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
novembar 1991. cena 120 din.

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17

Generalni direktor
Ilija Rapajić

Direktor Novinskog sektora
Zoran Nikodijević

Glavni i odgovorni urednik
Jovan Regasek

Zamenik gl. i odg. urednika
Esad Jakupović

Tehnički urednik / sekretar redakcije
Natalija Desančić

Urednik za hardver
Dejan V. Veselinović

Stručna redakcija
Vesna Ćosić (aktuelnosti), Nenad Ratočanin (baze podataka), Miodrag Lovrić (radne tabele), Dejan Ristanović (software), Jovan Skuljan (programiranje), Dejan Veselinović (hardver), Dragutin Vuković (mreže), Zoran Životić (software)

Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III

Telefoni
853-748 (redakcija)
850-528 (prodaja)

Agencija BIGZ-a
(011) 651-793, 653-565

Teleks
11655 BIGZ YU

Telefaks
(011) 648-140, 647-955

Pretplata za zemlju
Za jednu godinu 1080
Za šest meseci 540
(na žiro-račun D.P. BIGZ 60802-603-23264)

Pretplata za inostranstvo
Za jednu godinu 51 USD, 80 DEM, 87 CHS, 27 GBP, 271 FRF
(na devizni račun D.P. BIGZ 60811-620-16101-820701-999-03377)

Zbog usporenog prenošenja uplata preko banke molimo pretplatnike da nam posle svake nove uplate odmah pošalju foto-kopiju uplatnice

Rukopisi se ne vraćaju

SEZAM
sistem za modemske komunikacije (011) 648-959, 24 časa dnevno

Upravnik sistema
Jovan Regasek

Administratori na sistemu
Zoran Životić i Dejan Ristanović

Moderatori
Jovan Bulajić (PC USER), Aleksandar Damnjanović (PC OS), Slobodan Kalezić (ORKA), Vladimir Kostić, Vladimir Krstonošić (VICEVI), Esad Jakupović (RAČUNARI), Dalibor Lanik (ATARI ST), Pavle Peković (PC PROG), Dejan Ristanović (NOVOSTI i KOMUNIKACIJE), Bojan Tepavčević (FORUM), Dejan Veselinović (PC HARD) i Zoran Životić (SEZAM)

Sekretarica sistema
Natalija Desančić

Softver
Zoran Životić (glavna aplikacija) i Dejan Ristanović (eksterni programi)

Hardver
Server: COMPAQ DESKPRO 386/25, 3 MB RAM a tvrdi disk 300 MB (COMPAQ Minhen). Mreže: dve stanice COMPAQ 286N/12, 1 MB RAM a (COMPAQ Minhen), tri stanice „Ultra" AT 12 MHz, D-LINK DE 100 LanSmart („Ultre" Skopje). Modemi: VIVa (Thalema Communitations, Beograd)

SADRŽAJ

4 Novosti
ŠTA IMA NOVO

6 Računari / Epson 33
EPSONU NA ČAST

8 Procesori / i860
KRCKALICA ZA BROJEVE

10 Optimizacija hardvera
S GASOM DO DASKE

12 Tvrdi diskovi
TVRDA ČETVORKA

14 Video kartice
AKO KUPUJETE VGA

16 Video kartice
NOVI VGA ŠPIL

18 CAD / AutoCAD AME
PUNI MODELI

20 CAD / AutoCAD AEC
OD PODRUMA DO KROVA

22 Radne tabele / Lotus 2.3
LOTUS ZA SIROMAŠNE

25 Kompajleri / Clipper++
OBJEKTI NA KLIPERU

28 Programske biblioteke / Code Base
BAZE NA C-U

30 Stono izdaveštvo / Dizajn
RIZIK NESPRETNE UPOTREBE

32 Operativni sistemi / UNIX
SKELETNI SISTEM

35 Reply
KO JE ZALUTAO U CAD

36 Programski jezici / Ada
LAKI UVOD U ADU

38 Mreže / Trickle
SKUPLJANJE DATOTEKA

40 Tehnike programiranja / Asembler
HERKULES MEKOG SRCA

42 Tehnike programiranja / Windows
POPUNJAVANJE FORMULARA

44 Programski jezici / C++
APSTRAKTNE KLASE



Držeći se jednostavne maksime starih Latina "Omnia mea mecum porto" (Sva svoje sobom nosim), naša manekenka Maja Golić spremila je za putovanje pravu stvar: izvanredni računar EPSON 386/33. Na put (do naslovne strane) Maju je ispratila naša stručna ekipa: Irena Đurić (styling), Darija Miloš (make up) i Nenad Marjanović (fotografija).

49 Zoran Životić
MOJA ŠKOLA C-a

50 Savetnik / Turbo Pascal

52 Dejan Ristanović
BAJTOVI LIČNE PRIRODE

54 Dejanove pitalice
DEFEKTNE KUGLICE

Zbog teškoća u snabdevanju kvalitetnim papirom oktobarski broj „Računara" koji je trebalo da izađe iz štampe sredinom oktobra, pojaviće se na tržištu tek krajem oktobra / početkom novembra. Stoga smo bili prinuđeni da oktobarski broj preimenujemo u novembarski. Promenu datuma smo izvršili na koricama, ali je iz tehničkih razloga bilo nemoguće da to učinimo i u unutrašnjosti – u paginaciji lista. Pred vama je dakle, novembarski broj „Računara". Izvinjavamo se čitaocima zbog ove tehničke nepreciznosti. Ovaj pomak ćemo iskoristiti da se vratimo na stari datum izlaženja lista – na prvi u mesecu. Decembarski broj „Računara" dakle potražite na kioscima krajem novembra.

decembarski broj časopisa
računari
izlazi iz štampe krajem novembra 1991.

Na osnovu Mišljenja Sekretarijata za informacije SR Srbije broj 413-01-48 od 4. 2. 1991. na ovaj časopis plaća se osnovni porez na promet po povlašćenoj tarifi od 3 odsto.

MONITORI

MONITOR C21LV-65MAX

Pod imenom C21LV-65 MAX „krije se" kolor monitor dijagonale 21 inče Ovaj monitor, koji proizvodi korporacija *Image Systems*, radi sa horizontalnom frekvencijom od 15 do 65 KHZ i venkalnom od 55 do 90 KHz i može da radi sa ili bez prepitanje

Image Systems tvrdi da se sa ovim monitorom može dobiti slike u rezoluciji većoj od 1280 • 1024 Cena $4695

Kontakt adresa Image Systems Corp , 11543 K-Tel Dr , Hopkins, MN 55343, (612) 935-1171 fax (612) 935-1366

RAČUNARI

386SX/20 NOTEBOOK RAČUNARI

Na tržištu prenosnika vlada velike konkurencija Računari su sve manji po dimenzijama Konkurencija i napredak u tehnologiji donose sve snažnije računare u sve manjim „pakovenjima", uz sve niže cene

Korporacija *American Mitac* je objavila početak prodaje njihovog računara veličine sveske sa procesorom 386SX na 20 MHZ, pod imenom *3026E Workbook* Računar sadrži hard disk od 44 MB 2MB RAM-a (proširivo do 6MB u koracima po 2MB), 3 5" disketnu jedinicu i crno-beli VGA LCD displej, koji prikazuje 32 nijanse sive boje u rezoluciji 640 • 480 Računaru poseduje paralelni, serijski i interfejs za miša, priključke za monitor tastaturu i ispravljač/punjač Tastatura ima 82 tastera I sa nje su dostupna sve funkcije tastatura sa 101 testerom Isporučuje se uz DOS 4 01 Cena $2795

Kontakt adresa American Mitac Corp., 410 East Plumeria Dr , San Jose CA 95134 (408) 432 1160 fax (408) 432-0866

BOLJE I JEFTINIJE

Da može boje i jeftinije pokazuje firma *LSE Electronics*. Njihov 386SX NOTEBOOK računar radi na 20MHZ, ima ugrađen 120MB harddisk, male potrošnje izrađen od materijala na bazi stakla, 3 5" disketnu jedinicu, VGA LCD sa ugrađenim pozadinskim osvetljenjem i bateriju koja obezbeđuje ovom računaru neprekidno tri časa rada

Uz računar se dobija punjač i kutija za nošenje I DOS 4 01 Cene $2595

Kontakt adresa LSE Electronics, Inc , 77 West Nicholai St Hicksville NY 11801, (516) 931-1670, fax (516) 931-2565

BRAJOVA AZBUKA U DŽEPNOM RAČUNARU

Računari nisu nedostupni za ljude sa oštećenim čulom vida Pojavljuju se i džepni računari namenjeni slepima *BrailleMate* je džepni računar sa Brajovom tastaturom Ima ugrađen editor pomoću kojeg se podaci smeštaju u statičnu memoriju Sintetizator govora i elektronske Brajeve ćelija ugrađani u računar, potvrđuju unos sa tastature. *BrailleMate* može da uskladišti do 128 strana pisanih Brajevom azbukom Proširivanje memorije se vrši memorijskim karticama, veličine kreditne kartice Računar poseduje serijski i paralelni interfejs pa sa mogu priključiti Brajev ili običan štampač Cena $1595



Kontakt adresa TeleSensory, 455 North Bernardo Ave , Mountain View CA 94039 (415) 960-0920 fax (415) 969-9064

ŠTAMPAČI

LASERSKI ŠTAMPAČ ZA MREŽE

ALPS LSX1600 je naziv laserskog štampača dizajniranog za rad u mreži Brzine štampanje je 16 strane u minuti pri rezoluciji 300 • 300 Poseduje dve deveća hartije i ugrađeni davač za koverte, što omogućuje štampanje na dva papira različitih formata i koverti upućivanjem samo jednog zahteva

Kompatibilan je sa *LaserJet II* serijom štampača Opcionalno sa može ugraditi kertridž sa *PostScript*-om Poseduje 24 ugrađena fonta, s podržave i softverska fontove, kao i kertridže za *Hewlett-Packard* štampače Isporučavače se uz 1 MB memorije, prošinvo do 7 MB Paralelni i sanjski interfejs su standardno ugrađeni U davače za papir staje 750 listove Predviđeno puštanje u prodaju je u oktobru. Cene $3295

Kontakt adresa ALPS America, 3553 North First St , San Jose, CA 95134, (408) 432-6000 fax (408) 432-6035

ŠTAMPANJE NA DVE STRANE LISTA

Štampanje sa dve strane lista je standardna mogućnost *Xerox 4213* laserskog štampača Štampač je dizajniran tako da je olakšana izmena kertridže sa suvim mastilom razvijačem i fotoosetljivom supstancom Brzine štampanje iznosi 13 strana u minuti. Štampač podržave aplikacije pisane za daljinski upravljane računara i rad u računarskim mrežama Ugrađene je emulacije *Laserjet IID* serije i podržane su *Xerox Escape* sekvence Davači papira sa mogu kontrolisati softverski Cene $4795

Kontakt adresa Xerox Corp , P O Box 24, Rochester, NY 14692, (800) 332-2092

PERIFERIJE

CD ROM U KUĆNOJ RADINOSTI

Uređaj pod nazivom *The Personal RomMaker* služi kao sistem za backup podataka na digitalnoj traci i pripremu podataka za snimanje na kompakt-disk. Uređaj može snimati podatke u ISO 9660 ili *Hi-Sierra CD ROM* formatu. Snimljenu traku možete poslati u pogon za izradu CD-ROM-ova. U paketu se dobija sav potreban softver. SCSI interfejs za PC baziran na procesoru 386 i DAT (Digital Audio Tape) jedinica. Cena $7000

e je proizvod

Kontakt adresa JVC Information Products Co., of America 19800 Beach Blvd., Suite 1, Huntington Beach CA 92646. (714) 965 2610. fax (714) 968-9071

BEŽIČNI TRACKBALL

Bežični trackball, koji proizvodi *D1 International* uklanja još jedan kabl sa vašeg stola. Ovaj trackball je povezan sa računarom putem infracrvenih zraka. Kompatibilan je sa *Microsoft* i *Mouse Systems* mišem, nudi rezolucijo od 10 do 1200 tačaka po inču.

Proizvodi se u dve verzije, za Mac i PC računara. Cena $129.50

Kontakt adresa D1 International, 95 E Main St., Huntington, NY 11743. (516) 673-6666, fax (516) 673-6693

KOMUNIKACIJE

OPTIČKA KOMUNIKACIJA

Metropolitan Fiber Systems je najavio novi sistem za brzu komunikaciju putem optičkih vlakana, koji će omogućiti da računari, povezani preko urbanih optičkih magistrala razmenjuju podatke brzinom do 100 megabita u sekundi. U praksi sistem omogućuje proširenje FDDI mreže (*Fiber Distributed Data Interface*) izvan kompanija. Prenos podataka koji bi trajao 10 minuta uz upotrebu obične telefonske linije sa propusnom moći od 56 kilobita u sekundi, na FDDI se odvija za oko 3/4 sekundi.

Kontakt adresa Metropolitan Fiber Systems, One Tower Lane, Suite 1600 Oakbrook Terrace IL 60181, (708) 218-7200, fax (708) 218-0018

JEZICI

NAJAVA QUICK C ZA WINDOWS

Microsoft je zvanično najavio svoj C razvojni sistem za *Windows* okruženje. *QuickC for Windows* treba da omogući kraće i lakše pisanje *Windows* aplikacija. Kompanija tvrdi da grafička okolina sadrži sve alate potrebne za razvoj programa koji će raditi u *Windows* okruženju, kao i za konvertovanje DOS programe u programe koji će raditi pod *Windows*-ima.

U paketu se upotrebljava *QuickCASE W* tehnologija za generisanje koda, čiju je licencu *Microsoft* kupio od kompanije *Caseworks*.

Caseworks aleti su dizajnirani da olakšaju posao pravljenja grafičkog interfejsa tako što generišu kod za različite elemente ovog interfejsa („radio-dugmad" i ne primer).

QuickC for Windows je najavljen za 1. septembar. Preporučena maloprodajna cena je $199. Vlasnici *Microsoft C 5.0* paketa (ili novije verzije). *QuickC 2.0* kompajlera ili *QuickC*-a sa *QuickAssembler*-om 2.01 (ili novijom verzijom) mogu da izvrše doplatu od $89.95.

Kontakt adresa Microsoft, One Microsoft Way, Redmond WA 98052-6399. (206) 882-8080 (800) 426-9400

BORLANDOV ALAT ZA WINDOWS

Borland je izbacio na tržište svoj *Resource Workshop* alat za vizuelno kreiranje „resource" elemenata *Windows* grafičkog interfejsa kao što su ikone, dijalozi, fontovi, bez potrebe za pisanjem koda.

Workshop preuzima ulogu *Whitewater Resource Toolkit*-a koji *Borland* isporučuje uz *Borland C++* i *TurboRascal for Windows*. Borland kaže da će *Whitewater Resource Toolkit* nastaviti da se isporučuje uz ove proizvode dok se ne pojave nove verzije.

Resource Workshop je namenjen upotrebi uz *Borland*-ove i *Microsoft*-ove programske jezike. Omogućava modifikovanje *Windows resource*-a, konsteli vizualne *resource* editore, tekst editor za manipulaciju sa tekstualnim opisima *resource*-a i kompajlera koji omogućava postepeno kompajliranje *resource*-a.

Resource Workshop sadrži integrisani *resource manager* sa grafički orjentisanom bitmapom, dijalozima, fontovima, editorima menija. Cena $49.95

Kontakt adresa Borland International, Inc., 1800 Green Hills Rd. P.O. Box 660001, Scotts Valley, CA 95067. (800) 331-0877, (408) 438-8400

TRŽIŠTE

UNIXOVCI KORISTE MS-DOS

U anketi u kojoj je učestvovalo *800* članova *UniForum*-a, koji važi za internacionalno udruženje korisnika UNIX-a, 93.1% korisnika je navelo da koriste neku verziju UNIX-a, ali 81.1% još uvek koristi i DOS.

Što se tiče verzija operativnog sistema, većina korisnika je izjavila da koristi *UNIX SYSTEM V* (52.2%). *SCO UNIX* je na drugom mestu sa 37.7%, a ne trećem *SunOS* sa 35.2%. Zbir procenata je veći od *100* jer članovi koriste po više verzija operativnog sistema. Anketa je takođe pokazala da je broj korisnika *SCO UNIX*-a u značajnom porastu.

Što se tiče platformi, najrasprostranjenija je *IBM PC* sa 44.5%, na drugom mestu je *Sun* sa 38.6% i *Compaq* sa 38.1%.

Kontakt adresa UniForum, 2901 Tasmen Dr, Suite 201, Santa Clara CA 95054, (408) 986-8840

AMD TUŽI INTEL

AMD je podigao tužbu protiv Intel-a zbog monopolizacije tržišta uz odštetni zahtev od 2 milijarde dolara.

razvoj programa koji će raditi u *Windows* okruženju, kao i za konvertovanje DOS programe u programe koji će raditi pod *Windows*-ima.

Tužba je podnesena 29. avgusta u oblasnom sudu u San Jose-u u Kaliforniji, USA. Uz navod da je Intel prekršio federalni Šermanov i Klejtonov zakon protiv monopola, sprečavajući AMD da izađe na tržište sa 386 i 486 kompatibilnim čipovima. Tužba sadrži tvrdnje koje AMD navodi od 1987. uključujući optužbu da je Intel 1981. prekršio licencni sporazum sa AMD-om, da je Intel lagao AMD o svojim namerama u ugovoru i da Intel nastavlja se zastrašivanjem muštenje koje potražuju 386 i 486 čipove ako te muštenje posluju sa AMD-om.

Nova savezna tužba je samo najskorija u nizu legalnih akcija između AMD-a i Intel-a u zadnjih četiri godine – 1987. AMD je tužio Intel u Kalifornijskom Državnom sudu zbog povrede licencnog dogovora. Taj slučaj je bio poslat na sudsko razmetranje, mišljenje se očekuje do kraja 1991.

UMESTO KOMENTARA

NAMET NA VILAJET

Nije na kakost prvi put da se kao nas upotreba pameti službja administrativnim merama. Ili da se na nju odlazuju državni nameti. Primera računara koji nas na primer prvo je onemogućavanje potpunim zabranama uvoza, pa onda izazivolom supovine samo najjeftinijih (i najmanje korisnih) modela, a zatim onemogućavanjem uvoza štampača, drugih za rad veoma važnih komponenata. Da to besmislene i štetno nateranje države sa vlastitim potrebama za što brzom kompjuterizacijom u svim domenima nije bilo poučno pokazuje problem koji je ovih dana ulvoren a kome zbog opšte situacije u našoj zemlji dosad nija poklonjena pažnja koju nesumnjivo zaslužuje. Umesto hardvera, tema je ovog puta softver.

Problema su sa prve delivalli korisnici Saturna, krajem septembra, ovde je izlažeme poratečeno izvoda iz znatno šire rasprave u konferenciji PC SOFT (tema: Nelavlje). Verujući da oni čitavu stvar postavljaju dovoljno jasno, zbog čega ih objavljujemo bez posebnog komentara, ohrabrimo u toku rasprave i u „Računarima".

- „Savezna Uprava Carina je izdala obaveštenje, objavljeno u 'Carinskom pregledu' za jul/avgust 1991. pod brojem 65/91-A 1. prema kome će se obustaviti na programe za računare plaćati carina." (dejanvilovic, 29.08.91)
- „Nazdravlje! Ete nas još je nečemu prvih u svetu." (dejani, 29.08.91)
- „Stvari je definitivno konačna. Izgleda da je taj skrope donet dosta davno, ali se konačno tumačenje pojavilo tek nedavno. ( ) Intelektualni rad se obezvređuje direktno se radi protiv zakona o copyrightu, ohrabruje se domaći proizvođači softvera (specijalnog i knjigovodstvenog za sada). Mi smo u firmi jedva „drasnuli" kupce da kupuju legalno ovezen softver registruju je debijaju ugovarate mamo tiček da im poklanjamo uz računar softver koji nau plaćali. (zkehler, 01.09.91)
- „Za sad mogu da kažem samo evo: sam 'porez na softver izgleda da će biti 44% (20% + 24%, jer se oporezuje po dva osnova). ( ) Inače, projekt je donesen dosta davno, ali se tumačenje pojavilo tek u jednom od skorih izdanja nekakvog Službenika za.. (zkehler, 08.09.91)
- „Potrebam mi je tačan tekst tog člana 34, jer vodim kampanju (!) da 'Računari kao časopis urade nešto po tom pitanju, na primer, izraze protest vladi, obuve redakcijski uvodnik, pse petorje organizuju demonstracija (?) i slično. Možda izgleda uzaludne, ali nisi se ali ni se ne sviđa perspektiva da plaćam pare državi na takve stvari kao što je software. Još ako se neki ministari avale setl da naboje još neke dodatni poiez." (bulaja, 10.09.91)
- „Cini mi se da će carinici uskoro da naplaćuju i ono što imamo u glavama prilikom prelaska granice. Što znači: aha, bio si u Nemačkoj, gledao si blage, čitao (gledao) prospekte, naučio si nešto novo... plati carinu!" (jitov, 12.09.91)

U članu 34. Carinskog zakona definisa se da je carinska osnovica na koju se primenjuju stope carinske tarife vrednost uvezene robe, to jest govorena cena (transakcijska vrednost) odnosno stvarno plaćena cena ili cena koju treba platiti za robu kupljenu radi uvoza u Jugoslaviju (u članu se, inače, precizno objašnjave šta predstavlja ugovorena cena). U "Objašnjenju br. 65/91-A 1" Saveznе uprave carina ukazuje se da je prema objašnjenju iz marta ove godine u suštini dato ovajno tumačenje postupka utvrđivanje carinske osnovice za programe (softver). Prame jednom osnovicom je smatrana stvarna, takozvana transakcijska vrednost "medija" sa programom, po drugom vrednosti samog magnetskog medija koja je po pravilu skromna. Po novom objašnjenju medijum podrazno je prve tumačenje što formalno gledano ima smisla.

Problem ipak nije u tumačenju propisa nego u njegovoj suštini. Iz Objašnjenja se može izvući zaključak da je i SIV bio u dilemi i da je rešenje (tumačenje) preporučio rešenje samo kao prihvatljivu alternativu. Moglo bi se postaviti pitanje zašto je tumačenje dato baš (ili zašto tek) sada, sa zakašnjenjem od pola godine? Možda zbog "iskrenjenih" severnih fendova?

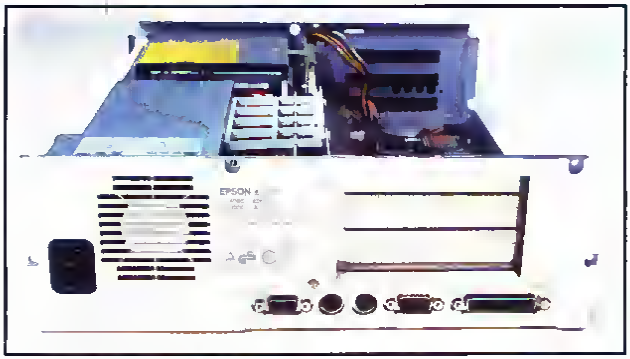
Ali to zapravo nije bitno. Mnoge je važnije pitanje: Zašlo uopšte namet na pamet? Možda je imanje previše?!

E. Jakupović

EPSONU NA ČAST

Odavno nismo imali prilike da se pozabavimo nekim od proizvoda poznate japanske firme EPSON. Iako je najpoznatija po svojim štampačima, i to pre svega matričnim, ova firma proizvodi i drugu računarsku opremu, a pre svega osnov – računara. Ovom prilikom ćemo pogledati model koji se nalazi blizu samog vrha piramide njihova ponude – EL3/33.

Dejan V. Veselinović



Veze sa spoljašnjim svetom. Epson EL3/33 od pozadi

Računar nije namenjen narodnim masama pre svega zbog cena koju valja podmiriti u slučaju takvog imena. Sem toga, računar se bazira na procesoru Intel 80386DX sa radnim taktom na 33 MHz i keš memorijom od 32 KB. U osnovnoj verziji mašina ima tvrdi disk kapaciteta od 120 MB i jednu flopi disk jedinicu prečnika 3,5 inča i kapaciteta 1,44 MB. Video podsistem je naravno saglasan VGA standardu ali pored osnovnog standarda podržava i tzv. Super VGA režima, sa rezolucijom od 800×600.

ESTETIKA

Prva specifičnost koju odmah primetite na računaru su njegove dimenzije: dubina je dosta standardna (389 mm), ali je visina veoma mala (99 mm), kao i širina (310 mm). Rekli bismo da je kućište čak veoma ukusno pa ako hoćete i lepo. Uprkos težini od oko 6,5 kg kućište je veoma solidno napravljeno od dobro obrađenog lima, pa se kulija veoma malo čuje prilikom prenosa.

Sa prednje strane se nalaze LED indikator za TURBO i DISK, prekidnici sa RESET i TURBO i najzad preklop za mrežni napon. Tu je i druga specifičnost računara – nema disk jedinice od 5,25 inča. Nema ni mesta za nju, zamenila ju je disk jedinica od 3,5 inča kapaciteta od 1,44 MB, baš kao i na IBM modelima iz serije PS/2. Nafik Compaq-u, i Epson se opredelio za specijalne, samo njima svojstvene priključnice za tastaturu i miša (PS/2 kompatibilnog). Drugim rečima, nema ništa od standardnih uređaja, manje-više sve vezani za Epson-a. Ovo nas nikada nije oduševljavalo, pa nije ni ovaj put.

Pošto miš ima sopstveni vezник, Epson se zadovoljio serijskom ugradnjom samo jedne RS-232C veze (9 igličnі vezник) i jedne paralelne veze. Sve to je već ugrađeno, odnosno nema potrebe dokupljivati te delove odnosno kartice (ugrađeno je na matičnu ploču).

Tastatura je doste neobična. I podeća nas po osećaju na Keytronics tastature. Potpuno je lična, osrednje mekana ili tvrda, kako hoćete, ali ima pozitivan osećaj davanja kontakta. Drugim rečima, nemate onaj odvratan osećaj propadanja prstiju u bezdan: tačno znate kada ste pritisnuli taster.

TEHNIKA

Epson nikada nije bila sastavljačka firma, to znači da su sami oduvek projektovali i proizvodili delove za sopstvene mašine, delove koje su istine, pravili drugi u njihovo ime i za njihov račun. Do sada, Epson nije pokazivao neke veće želje da se istakne nekom tehničkom izuzetnošću ili novotarijom, ali sada su se vremena očigledno veoma promenila.

Osnovna koncepcija konstruktora je veoma interesantna, jer nudi neka zaista jedinstvena rešenja. Matična ploča sa sastoji od zapravo četiri pločice. Dve su položene horizontalno i međusobno povezane jednim posebnim vezником (igle/utičnice), ne jednoj od tih pločica se nalazi procesor sa keš kontrolerom i nekim vezним VLSI kolima bitno vezanim za sam procesor, a na drugoj (horizontalnoj) ploči) se nalaze sva veza sa sabirnicom i ostala logika (CMOS, BIOS, AT Bus adapter, flopi kontroler, paralelna i serijska veza) koja nije odrađena samim procesorom (BIOS je u obliku EPROMA, pa je stoga lako zamenljiv).

Jedna od vertikalnih pločica sadrži tri 16-bitne utičnice za kartice, koje su prazne u osnovnoj konfiguraciji, a na drugoj vertikalnoj pločici se nalazi video podsistem. Pošto se te dopunske pločice postavljaju paralelno sa dnom računara, kućište je izuzetno usko i nisko, dok je dubina dosta standardna.



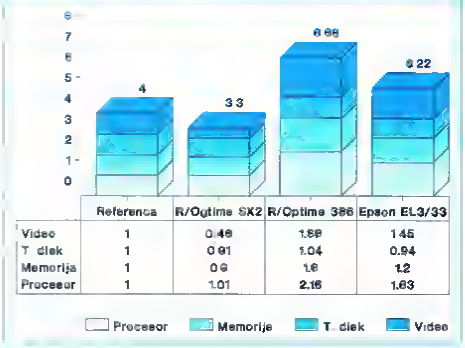
Otvor u malom pakovanju. Epson EL3/33 spreda

ELEKTRONIKA

Sva elektronika je u realizovana u VLSI kolima sa očigledno visokim stepenom integracije. Pored procesora Intel 80386DX 33. Tu je i posebno za odgovarajući koprocesor. 6 čipova sa oznakama firma Epson, jedno sa oznakom firme Toshiba, a ostali su prilično standardna TTL i PAL kola.

Video podsistem pobuđuje jedno Western Digital WD90C10 LR (isto kolo koje pobuđuje razna Paradise karte). Sva video memorija je skoncentrisana u dva ZIP kola sa oznakom

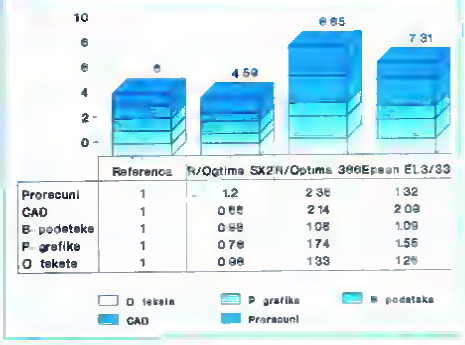
Slika 1 Epson EL3/33



	Reference	R/Optima SX2	R/Optima 386	Epson EL3/33
Video	1	0.49	1.59	1.45
T. disk	1	0.91	1.04	0.94
Memorija	1	0.9	1.6	1.2
Procesor	1	1.01	2.16	1.63

Mešinski testovi

Slika 2 Epson EL3/33



	Reference	R/Optima SX2	R/Optima 386	Epson EL3/33
Proracuni	1	1.2	2.36	1.32
CAD	1	0.66	2.14	2.09
B. podataka	1	0.99	1.08	1.09
P. grafika	1	0.76	1.74	1.55
O. teksta	1	0.96	1.33	1.25

Programski testovi

514256 koji zajedno zamenjuju osam standardnih DRAM čipova za ukupan komplement memorije od 256 KB.

Interesantno je i rešenje sistemske memorije. Radi se o specijalnim SIM modulima. Čipovi se nalaze sa obe strane modula sa svake strane se nalazi po osam čipova sa oznakom 514256-8 (4×256 KB, 80 nS) i po četiri čipa od po 256 KB (parite). Tako svaka strana modula sadrži po 1 MB, odnosno 2 MB za ceo modul. Pošto postoje tri postolja, ukupna količina memorije iznosi 6 MB, a eko bi se ovi moduli zamenili drugima, ukupna količina bi mogla iznositi i celih 24 MB. Na žalost, u priručniku se navodi maksimum od 12 MB, no, pošto se radi o računaru koji očigledno nije namenjen ulozi servera, već konsnicima-pojedincima, to i nije neka posebna mana.

Sve je čak izuzetno uredno, kao što se na slici može videti, na ovakvoj urednosti bi Epsonu mogli pozavideti mnogi drugi, jednako poznati proizvođači.

OPREMA I OBRADA

Obrada računara je na najvišem nivou, onako kako bista i očekivali od velikog svetskog imena. Od odlično obrađenih spoljnih površina preko dobrog dizajna do uredno složenih pločica i veznih kablova iznutra. Ala! im vele.

Pored toga, uz računar dobijate i zaista odlično pripremljen paket priručnika. Dva se odnose na DOS 4.01 koji se dobija uz računar, jedan je sam računar, a poslednji na bzio podešavanje parametara mašine. Priručnici su odlično pripremljeni i prezentirani, ali što je daleko važnije, izuzetno su informativni. I ovde Epson mirne duše beleži pun pogodak.

Naravno, u paketu dobijate i sve potrebne diskete – tri se odnose na DOS, jedna na sam računar („Reference disk") a poslednja sadrži razne veznike za video podsistem. Interesantno je primetiti da veznika ima veoma različitih, neki su za već poodavno zastarele programe (recimo „Framework II"), neki imaju razne nivoe (WordPerfect 4.1, 5.0 i 5.1), a neki su za dosta napredne sisteme (Epson OS/2) „Auto-CAD" V 11 nije podržan.

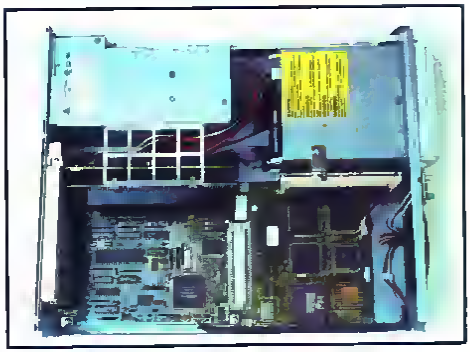
MERENJA

Merenja su pokazala dosta čudne rezultate. U nekim slučajevima, EL3 je veoma osetno podbacio u odnosu na konkurenciju, u drugima je bio na nivou, a u trećima je iznenadio rezultatima. Recimo, u procesorskim testovima je zaostao za našom R/Optimom 386 Mk 3 za celih 30,7%, što je do sada najveća razlika na koju smo naišli u okviru iste klase mašina, a sa druge strane, u testu se obradom tekstova pokazao se sponjim od R/Optime za svega 5,5%. Ovakvo šetanje razultate ipak nismo verovali da je moguće, ali eto, i to se desilo.

U celini, u mašinskim testovima Epson se pokazao nekih 27,7% sponjim od naše R/Optime 386 Mk 3 (kojoj je ovo, uzgred budi rečeno, poslednje gostovanje u javnosti jer je zamenjena novim poslovnim standardom R/Optimom 486 Mk 2), što je funkcija lošijih rezultata na testovima procesora i video podsistema.

Sa druge strane, u slučaju programskih testova Epson se ponašao sponje od naše reference (-17,2 +1%), pa pošto programske testove smatramo metodavnijim možemo reći da se radi o mašinama sa razlikom u brzini od prosečno 15%. Razliku koja objektivno postoji garantovano nećete osetiti u svakodnevnom radu, ako računar koristite za manje poslove, ali hoćete eko nameravate da se bavite CAD/CAM-om ili DTP-om.

Merenja su ukazala na veoma biz tvrdi disk, ali i na dosta spor video podsistem. Ovo je pomalo zaćudujuće, s obzirom da se proizvođač opredelio za sopstveni razvoj, kade se zna da se ovim dužim ekupljim i komplikovenijim putem najčešće ide kako zbog kvaliteta, tako i zbog boljih performansi. Prosto šteta, jer imamo nekakav osećaj de posao nisu sasvim



Odličan rad: Epson EL3/33 iznutra

završili, kao da su umorili i stali pred samim ciljem. Za utehu nisu jedini to se i IBM-u često dešava.

ZAKLJUČAK

Epson EL3/33 je jedna pomalo neobična mašina i spolja i iznutra, u ovom slučaju neobičnost ima pozitivan predznak. Izuzetno je simpatičan na oko, odlično obrađen i spolja i iznutra, iako se ovo drugo ne vidi u svakodnevnom radu. Na žalost, dobru nameru nisu doveli do potpunog kraja odlično napravljen računar i dobro odabrane jedinice spoljne memorije, ipak ostaju prikraćene za malo brži i jači video podsistem.

Iako se bojimo de će cena ovog računara biti veoma velika, moramo priznati de nam se dopao i da bismo ga rado zadržali de su nam dozvolili. Nedostatak disk jedinice od 5.25 inča, nestandardna priključnica za tastaturu, svega tri mesta za proširenje specijalni memorijski moduli, sve su to relativne mane, ali je celina ipak takva da je te mane veoma lako zaboraviti.

Ono što i posle oduzimanje poena ostaje je i dalje za svaku pohvalu. Proizvod koji služi na čast svom proizvođaču i samo potvrđuje njegov slečeni zavidni renome.

Kontakt adresa

Preduzeće MZ
Dubljanska 70, YU-11000 Beograd
Tel. (011) 434-812, Fax (011) 450-471

KRCKALICA ZA BROJEVE

Postoji brzo, postoji još brže, ali nema najbržeg; današnja munja je sutrašnji sporac. Procesor i80XX je bio za ono vreme brz, ali sa koprocesorom je bio još brži; procesor i486 je već dogurao do nivoa mini računara, a koprocesor mu je ugraden. Ni to nije dosta, misli američka firma „Microway" i predstavlja svoj model „matematičke maašine" sa RISC procesorom i860.

Bojan Topavčević

I pored strahovitog razvoje personalnih računara, koji se po performansama sve više približavaju velikim sistemima, za neke namene brzine jednostavno nikad nije dosta. To se, pre svega, odnosi na razne vidove animacija, 3D grafiku i naročito na intenzivne numeričke proračune. U želji da slakne predstavu o realnim brzinama, pisac ovih redova je kroz duži period vremena ispitivao koliko je vremena potrebno različitim platformama personalnih računara da reše određene sisteme linearnih jednačina (takozvani LINPACK test). Rešavanje sistema linearnih jednacina je veoma često i sa aspekta brzine, to je neuralgicna tačke za razne vrste proračuna, pa su podaci dobijani ovakvim testovima prilično relevantan pokazatelj brzina za poslove sa takvom namenom.

UBRZANA NUMERIKA

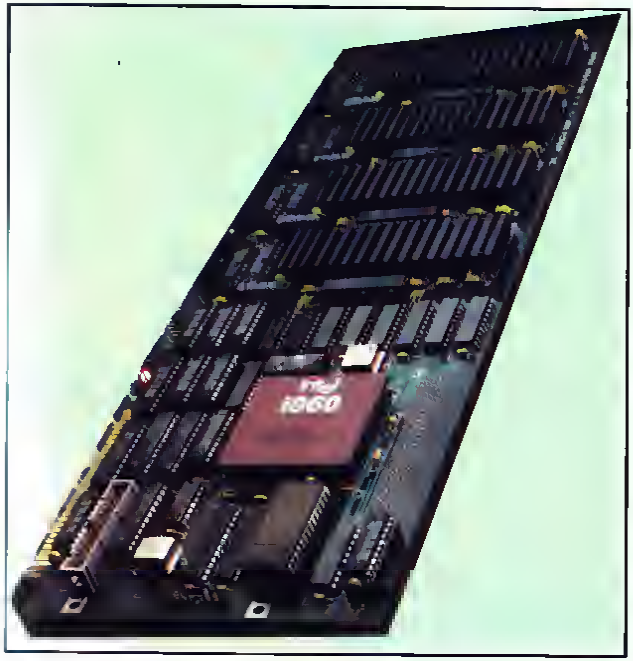
U svetu postoje firme koje su specijalizovane ze proizvodnju hardvera i softvera (za PC računare) koji ubrzavaju numerička izračunavanja. Jedna od najpoznatijih je firma „Microway (iz Kingstona, Massačusets, SAD) koja za sebe tvrdi da je „World leader in PC numerics" („Svetski vođa u oblasti numerike"). Od same pojave Abacus-ovih Weitek alternativnih koprocesora, firma Microway ih je softverski podržavala raznim kompajlerima. Pored toga, firma je uz odgovarajuću softversku podršku nudila kartice bazirana na Inmosovim T800 transpjuterima (MonoPuter, BiPuter, QuadPuter i Hex Puter sa 1, 2, 4 ili 6 transpjutera), koje su u doba 16 i 20 megaherecnih 386-ca donosile značajna ubrzanja. U poslednje vreme glavni Irkacki konj firme „Microway" bez dilame je kartica „Number Smasher 860" bazirana na Intelovom RISC procesoru i860. Zahvaljujući ljubaznosti zaposlenih u institutu „Boris Kidrič" iz Vinče, autor je imao priliku da testira pomenutu karticu.

Karta se standardno prodaja u verzijama na 33MHz ($7300) i 40MHz ($9200), sa 8MB četvorostruko isprepletana (?) 100ns RAM memorije. U cenu karte uračunat je i jedan kompajler (po izboru C, Fortran ili Pascal).

Karta se priključuje pomoću standardnog 16-bitnog ISA slota preko kojeg sa računarom ostvaruje komunikaciju od 1.5 MB u sekundi. Opcionim 64-bitnim FIFO interfejsom moguce je ostvariti komunikaciju od 4MB u sekundi preko ISA slota ($500) ili 32MB/sec preko EISA slota ($1000). RAM memorija se može proširiti do 32MB ($4000). Preko 64-bitnog 860 bus-a za podatke moguće je ovu kartu uz brzinu od 60MB/sec povezati se drugim 860 karlama ili transpjuterskim kartama i uz podršku odgovarjućeg softvera ostvariti paralelnu obradu.

Prema proizvođaču karta postiže 80 miliona floating point operacija u sekundi u običnoj tačnosti i 10 Linpack MFLOPS-a u dvostrukoj tačnosti, kao i preko 500 000 vektorskih transformacija u sekundi.

Kompajleri koji podržavaju ovu kartu su NDP C-860 (sa punom BSD Unix i ANSI implementacijom) NDP Fortran 860 (sa FORTRAN 77 + ANSI X3.9-1978, VMS, BSD 4.2, FORTAN 66 i MIL-STD-1753 podržanim standardima i ekstenzijama) i NDP Pascal 860 (sa ANSI/IEEE X3.97-1983, BSD 4.2 i BSI sandardima i ekstenzijama). Cena kompajlera je 1995$ po komadu.



SUMNJIVA KOMPATIBILNOST

Testirana je varzija karta na 33MHz sa 8MB RAM-a uz koju je bio isporučen NDP Fortran 860. Testiranje se sastojli od merenja vremena potrebnog za rešavanje sistema određenog broja linearnih jednačina. Sama matrica sistema je simetrične, a elementi matrice koji su različiti od nule su grupisani oko glavne dijagonale. Rastojanje najudaljenijeg elementa različitog od nule od glavne dijagonale obično se naziva BAND. Sam test se sastoji od tri dela. Test 1 – rešvanje 558 linearnih jednačina sa 558 nepoznatih i BAND-om 100 (za ovaj test dovoljno je 640K RAM memorije tako de ga je moguce sprovesti na skoro svakom PC računaru). Test 2 – 885 jednačina sa BAND-om 500 (potrabno 4MB RAM-a) i Tasl 3 – 2081 jednačina sa BAND-om 500 (potrebno 8MB RAM-a). Pored brzine testirana je i preciznost platforme koja se ogleda u dva pokazatelja maksimalnoj greši (greška se odbija lako što se elementi matrice pomnoze sa rašanjima i dobijeni rezultat oduzme od slobonog člana što bi teoriski trebio da daja nulu).

I pored vrlo bombastičnih navoda u reklamama o velikoj kompatibilnosti i mnogim podržanim ekstenzijama, autor je ipak naišao na prilčan broj problema prilagodavajući izvorni kod testi programa za ovu platformu. I pored dosta visoke verzije kompajlera (4.0a), stice se utisak da će proizvođač morati još mnogo na njemu da poradi. Autor nije imao vremena da temeljnije analizira bagove ovog kompajlera, a najdrastičnije od onoga što je uspeo da lokalizuje je pre svega nesnalaznja sa EQUIVALENCE naredbama i greške pri prenosu parameta-

Tabela 1: 568 jednačina, BAND 100

	Platforma	sec	index	maxErr	avrEr
1	Microway i860/33	1 32	11 318	6 23E-12	7 48E-13
2	Opti 486/25 128K cache (protected mode)	5 65	2 644	3 91E-12	4.70E-13
3	486/33 256K cache, 3ws (protected mode)	6 15	2 429	3 91E-12	4.70E-13
3 w	486/33 256K cache, 3ws + Weitek 4167/33	7 03	2 125	4 83E-12	7.39E-13
3 r	486/33 256K cache, 3ws	8 56	1 745	3 91E-12	4.70E-13
2 r	Opti 486/25 128K cache	11 48	1 301	3.91E-12	4.70E-13
4	DTK 386/33 + Cyrix 3D87/33 (prot.mode)	14.77	1 012	3 91E-12	4.70E-13
5	386/33 + Cyrix 3D87/33 (protected mode)	14 94	1 000	3 91E-12	4.70E-13
6 w	Everex 386/20 + Weitek 1167/20 (pr mode)	17 47	0 855	5 45E-12	6.31E-13
7 w	Gateway 386/25 + Weitek 3167/25	19 11	0 782	5 45E-12	6.31E-13
4 r	DTK 386/33 + Cyrix 3D87/33	20 93	0 714	3 91E-12	4.70E-13
5 r	386/33 + Cyrix 3D87/33	22 13	0 675	3 91E-12	4.70E-13
8	386/25 + IIT 3C87/25 (protected mode)	23 07	0 631	4 62E-12	5.90E-13
6	Everex 386/20 + 387/20 (protected mode)	29 60	0 505	3 91E-12	4.70E-13
8 r	386/25 + IIT 3C87/25	35 98	0 415	4 62E-12	5.90E-13
6 r	Everex 386/20 + 387/20	39 76	0 376	3 91E-12	4.70E-13
9	DTK 386SX/16 + Intel 387SX/16 (p mode)	51 14	0 292	3 91E-12	4.70E-13
10 r	386SX/20 + IIT 3C87SX/20	56 24	0 266	4 62E-12	5.90E-13
9 r	DTK 386SX/16 + Intel 387SX/16	70 47	0 212	3 91E-12	4.70E-13
11	MicroVax II	74 11	0 202	5 89E-13	1.22E-13
12	286/20 Laptop + IIT 2C87/12	94 25	0 159	4 62E-12	5.90E-13
13	286/12 + Intel 287/8	219 82	0.068	3 91E-12	4.70E-13
14 q	MicroVax II (Quad precision – 128bit)	4382.58	0.008	5.89E-13	1.22E-13

TABELA 2: 885 jednačina, BAND 500

	Platforma	sec	index	maxErr	avrErr
1	Microway i860/33	33 71	11 368	1 23E-11	1.39E-12
2	Opti 486/25 128K cache	146 60	2 619	1 14E 11	1.39E-12
3	486/33 256K cache 3ws	181 42	2 116	1 14E 11	1.39E-12
3 w	486/33 256K cache 3ws + Weitek 4167/33	194 05	1 978	6 18E 11	1.39E-12
5	386/33 + Cyrix 3D87/33	383 86	1 000	1 14E 11	1.39E-12
4	DTK 386/33 + Cyrix 3D87/33	390 63	0 983	1 14E 11	1.39E-12
7	Gateway 386/25 + Weitek 3167/25	468 02	0 820	1 18E-11	1.39E-12
6	Everex 386/20 + Weitek 1167/20	491 75	0 781	1 18E-11	1.39E-12
8	386/25 + IIT 3C87/25	583 80	0 658	1.32E-11	1.39E-12
6	Everex 386/20 + 387/20	795.54	0.483	1.18E-11	1.39E-12

TABELA 3: 2001 jednačina, BAND 500

	Platforma	sec	index	maxErr	avrErr
1	Microway i860/33	106 83	11 337	7 97E 10	7 47E-11
2	Opti 486/25 128K cache	458 24	2 643	8 23E 10	8 10E-11
3	486/33 256K cache 3ws	578 20	2 095	8 23E-10	8 10E-11
3 w	486/33 256K cache, 3ws + Weitek 4167/33	618 16	1 966	6 25E 10	6.86E-11
5	386/33 + Cyrix 3D87/33	1211 16	1 000	8 23E-10	8.10E-11
8	386/25 + IIT 3C87/25	1845.11	0.656	1.04E-9	1.02E-10

ne u potprograme. Ovo je veoma začudilo autora s obzirom de je izvorni kod test programa bez ikakvih probleme prolazio kod „stanjeg brata" ovog fortrana NDP Fortrana-386 (kao i kod mnogih drugih kompajlera obično drastično jefinijih)

Sa druge strane dokumentacija za samu kartu je veoma štura a dokumentacija za kompajler je više nego loša. Naime proizvodač uz ovaj kompajler isporučuje odkumentaciju za NDP Fortran-386 s tim da poziv za razne specifične rutine (vreme datum i slično) koji su navedeni u uputstvu jenostavno nisu podržani, e podržani su neki drugi koji u upuststvu nisu dokumentovani (i da ne bi nešto demo izvornog koda na disketama teško da bi se uopšte za njih tako lako moglo saznati). Međutim i pored navedenih problema posle malo mučenja stvari su lpak došle na svoje mesto i „Number Smasher-860" se pokazao u punoj snazi. U odnosu na 486/25 sa 128K drugostepenog keša ova karte je brža 4 3 puta. Ostaje da se vidi de li se isplati izdvajati dopunskih $7200 za takvo ubrzanje. Po mišljenju autora za onoga koma je brzina preko potrebna, svakako se isplati

NA UPOREDNOM TESTU

Porad ove karte ne identičan način je kroz duži vremenski period, testiran veći broj različitih računara e rezultati su takoda priloženi u tabelama. Test program je pisan u fortranu u dvostrukoj tačnosti. Verzija test programa za AT računare, kao i za 386 i 486 računare u realnom rezimu dobijena je pomoću Microsoft ovog Fortrana 5.00. Verzije test programa za 386 i 486 računare u zaštićenom 32 bitnom modu dobijena su pomoću MicroWay NDP Fortrena 386 1.4e. Svi testovi računara koji imaju Witek koprocesor dobijeni su istim kompajlerom. Kod MicroVax-a II korišćen je VAX Fortran 77. Za „Number Smasher 860" korišćen je naravno MicroWay NDP Fortran-860

Potrebno je napomenuti da PC računari sa 8MB RAM-a koji imaju popunjene memorijske banke rade brže pri radu sa većim količinama memorije. To objašnjava pojavu da recimo računar sa matičnom pločom DTK 386/33 (4) koja je objektivno brža od ploče 386/33 (5) nepoznatog proizvođača sa Tajvana (uzete za reper) radi sporije od ploče 5 na testu 2 za razliku od testa 1. Sa druge strane testirani računer 486/33 (koji je proizveden u SAD) ima 32MB RAM memorija ali je ta memorija po svoj prilici veoma spora pa je računar morao de radi sa 3 stanje čekenja pri čitanju i jednim stanjem čekanje pri pisanju u memoriju pa je zbog toga postizao lošе rezultate. Iz BIOS-a je bilo moguce smanjiti broj stanja čekanja i tada je računer znatno brže radio ali sa greškom koje je bila za 1010 veca od standardne.

Sa druge strane, ovaj računer je bio veoma interesantan za autora zato što je to bila prva 486-ica sa Weitek 4157 koprocesorom do koje je uspeo da dode. Naime procesor i80486 sa integrisanim koprocesorom „487" je u brzini floating-point operacija napravio mnogo veci skok u odosu na razliku u brzinama između samih 386 i 486 procesora pa se opravdano postavilo pitanja da li se kupovinom alternativnog Weitek-ovog koprocesora (koji nije nimalo jeftin 4176/33 košta $1495) u toj kombinaciji išta dobije u brzini. Po ovim testovima je ispalo da kombinacija 486+"487" radi 6-10% brže od kombinacije 486+4167. Medutim imajući u vidu de Weitek-ovi koprocesori komuniciraju sa procesorom preko memorije a uzimajući u obzir da konkretni računar nije imeo baš najbržu memoriju, dobijena rezlutata za kombinaciju 486+4167 treba uzeti sa rezervom bar dok se stvari do kraja na ispitaju.



Dakle kao što se iz priloženih rezultata može videti ako vam treba veoma brza platforma za ovakvu vrstu numeričke obrade podataka, Microway „Number Smasher i860" je zaista dobro rešenje, mada naravno i dosta skupo. Jedina olaksavajuća okolnost je što za ovu karticu možete uzeti jeftinu podlogu u vidu nekog jednostavnijeg i proporcionalno jeftinijeg računara, tako odlazimo od odnosa od oko 2 puta veca cene kombinacija AT mašine i pločice „Number Smasher 860" u odnosu na recimo 486 na 25 MHz. U takvom odnosu placate duplo ali dobijate bar tri puta vecu brzinu proračuna, pa ako vam treba i ovako gledano se isplati.

S GASOM DO DASKE

O raznim programima i metodama optimizovanja rada mašinske opreme smo već više puta govorili na stranicama „Računara". Pa ipak, izgleda da tu ipak ostaje dosta prostora za razna dopunska objašnjanja i razmatranja. Evo nekih od njih.

Dejan V. Veselinović

Da je poenta rada sa računarima u tome da uradimo što više za što kraće vreme i da posao obavljamo što komfornije, nema nikakve sumnje. Izgleda da jedino preostaju neke nedoumice oko toga kako optimizovati računar i što je još važnije, kako na najbolji mogući način upotrebiti posložene programe za optimizaciju. Ovo drugo se poslavlja kao relativan problem, uprkos činjenici da se čitav niz takvih programa instalira pomoću automatskih potprograma za instalaciju, koji bi trebalo da sami pronađe neke „optimalne" parametre.

Ne želeći to se prilično retko dešava. Razloga za to ima nekoliko, a prvi je svakako veliko šarenilo mašinske opreme sa kojom se takvi programi susreću što dovodi do izbora prosečnih a ne optimalnih parametara podešavanja. Čak i kod takvih programa koji imaju veoma razvijene mogućnosti samopodešavanja, kao što je, recimo, *PC Kwik*.

Pored toga, postoji i nekoliko pitanja na koja sam korisnik mora da odgovori pre svega samom sebi, jer program ne može znati tačnu konfiguraciju računara i penterija, posebno onih van računara (koje po definiciji ne vidi i ne može videti).

OPTIMIZACIJA MAŠINSKE BAZE

Prvi korak ka optimizaciji je svakako u samoj mašinskoj bazi, odnosno unutar samog računara i bez ikakve spoljne pomoći. Navešćemo neke tipične mogućnosti i naša iskustva sa njima.

SHADOW RAM – BIOS. Radi se o prebacivanju sadržaja sporih (150-200 nS) 16-bitnih ROM čipova u brže 32-bitne memorijske segmente na istim adresama, nakon čega se ROM praktično isključuje iz daljeg rada, odnosno prestaje da postoji u funkcionalnom smislu jer se za te rutine računar dalje obraća brzom RAM-u. Brzina čitanja a samim tim i izvršavanja ovih rutina se skraćuje na brzinu pristupa RAM-u koja može biti tako velika da zahteva memoriju sa brzinom od čak 15 nS (u slučaju procesora 80486 na 50 MHz i reda bez stanja čekanja posredstvom keš memorije).

Nikada nismo videli bilo kakavo obrazloženje odnosno prikaz rezultata koji se dobijaju upotrebom ove dupli kat ne memorije. Uredili smo niz testova u želji da nađemo neka vidljivija rezultate, ali bez uspeha. Jednostavno, današnje stanje tehnologije je došlo dotle da ova pogodnost nema opipljivih praktičnih rezultata. Danas je sve mnogo brže no što je bilo pre 4 ili 5 godina, kada je ova mogućnost možda nešto i značila, ali danas izgleda više nema bitnijih dobiti od nje.

Sa druge strane, veliki (najveći) deo matičnih ploča ovaj segment automatski i neopozivo zauzima čim u računaru ima više od 1 MB RAM-a, pa dilema upotrebiti je ili ne ostaje samo onima sa 1 MB i mogućnosti da na ovo utiču, jer ne nude sve ploče i svi BIOS-i mogućnosti prebacivanja segmenta memorije od 640 KB od 1 MB na granicu iznad 1 MB. Ako nude, kao recimo pakeli čipova od firmi kao što su C&T ili OPTI, sa BIOS-ima firmi AMI i Award, učinite to, prebacite tu memoriju iznad granice od 1 MB i upotrebite je za keširenje tvrdog diska. Time dobijete neuporedivo opipljivije rezultate nego bilo kako drugačije.

SHADOW RAM – VIDEO BIOS. U principu, radi se o potpuno istom efektu, ali sa BIOS-om ne video kartame (32 KB od segmenta C000h). Ne ovom mestu se nalazi program predviđen za EGA i VGA kartice, pa će ovo imati efekta jedino sa njima, ali ne i sa CGA i Hercules karticama. Isto tako ubrzanje će biti čak veoma veliko, kreće se od 2:1 na najsporijim AT mašinama (80286 na 8 MHz, 1 stanja čekanja) pa sve do 7:1 na brzim mašinama sa procesorom 80486.

No pazite, sva ta ubrzanja se odnose isključivo na teksualni režim rada i sa grafikom nemaju nikakva veze. Drugim rečima grafički prikaz će ostati jednako brz u obe slučaja sa ili bez keširanja video rutina. Drugi problem na koji možete naleteti je da vaš prikaz teksta postane prebrz, verovali ili ne, ovo ume da bude čak i veći problem od presporog prikaza. Kod sporog prikaza jedini problem je čekanje, a kod prebrzog morate će se sa vraćate unazad i da se mučite oko locirania mesta do koga biste želeli da slignala. Istine neki programi (recimo, *WordPerfect 5.1*) vam nude mogućnosti da u njima regulišete brzinu skrolovanje bez obzira na računar, pa ih isto logikom možala upotrebiti i za usporavanje (!) prikaza eko je baš prebrz.

Za sve vlasnike AT mašina (sa procesorom 286) kao i vlasnike računara sa procesorom 386SX, preporučujemo da ovu mogućnost isključite, osim ako razmisle, a u svakom slučaju, neka probaju.

AT BUS SPEED. Ovde ima najviše nedoumica često i zbog zelene nemuštoge prikazivanja ove mogućnosti od strane proizvođača ploča i/ili autora BIOS-a. Radi se o pokušaju prevazilaženja problema koji mogu nastali kada je procesor jednostavno prebrz za standardnu sabirnicu (primer: 80486) odnosno kada su kartice na sabirnici ili starije pa ne trpe brzine, ili lošeg kvaliteta (ne žalost, sve češći slučaj), odnosno kada u računar strpate proizvode vrhunskog kvaliteta koji mirne duše podnose i daleko veće brzine od standardnih. U prvom slučaju cela možda imati potrebe da sabirnicu usporite, a u drugom da je eventualno ubrzala.

Standrdan radni režim sabirnice je 8 MHz sa jednim stanjem čekanja, to i samo to je prva ISA (Industry Standard Archiecture) sabirnica. Ova sabirnica vodi poreklo od IBM ovog modela AT računara druga generacije, kada je i procesor radio pod istovetnim uslovima. Danes međutim nije ni malo lako sinhronizovati takvu sadirnicu sa procesorom 386 ili 486 na 25 ili više MHz radnog takta, jer se radi o brzinama bar 8 puta većim od onih koje je dobri, stari 80286 mogao da izvuče u onoj verziji. Usaglasiti cikluse procesore sa sabirnicom postaje sve veći problem, koji se na korisnike odražava tako što im razne karte ili ne rade, ili rade ali loše i nepouzdano.

Prva i osnovne mogucnost jeste de dobro odaberete i odrešite kesu prilikom izbore delova ukoliko ete uopšte u mogućnosti da ne to utičete. Primera radi, AT Bus kontrolera ima u Minhenu već od DEM 25 nagore, nakon dosta probanje mi konstimo jedan takav kontroler sa oznakom „Microscience" (prodaje se i u verziji ne kojoj piše „Maxtor") koji košta tačno tri puta toliko, ali bez ikakvih problema trpi brzine od 12.5 MHz na sabirnici (radi se o kartici koje je samo kontroler, dakle nema ni jednu I/O vezu na sebi, to je ne posebnoj kartici). Na žaloal većina prodaveaca kod nas e i u Minhenu, ni ne drži takve kartice u nekim rasponima, već u jednoj jedinoj varijanti, eko je dobro odabrana nema problema, ali ako je osetljiva na brzinu računara, problema će biti.

Primera radi, današnje veoma male (tačno širine 16-bitnog konektora) AT Bus kartice neće raditi na mašinama sa procesorom 80486 ukoliko ne usporite sabirnicu, isto važi i za sve kombinovane (AT Bus plus I/O vaze) kartice koje smo do sada videli. Problem prvo očekujte na čitanju slanih disketa od 360 KB, kod kojih je brzina prenosa veoma mala, sistem će se pobuniti kao da nema diskete ili kao da ona nije u radu. U goram slučaju, sistem će sporadično imati problema sa pristupom disku, a u nazgodnijoj varijanti, počeće da se pojavljuju loši sektori, klasteri će se razigrali po disku i slično. Ako se ovo dogodi, sasvim sigurno imate probleme sa kontrolerom – ili je „crkao", ili ne može da se usaglasi sa procesorom. Često će se ponašali veoma sporadično – nekada će sve biti u redu a nekada ne.

Prvo pokušajte da smirite malo sabirnicu. Ako imate C&T kola i AMI BIOS, prebacite CLKIN na „AT BUS" ili „CLKIN/X" pri čemu „x" označave podelu radnog takta sa datim brojem (recimo, za takt od 25 MHz i x=2, sabirnica radi na 12.5 MHz, prebacite ja na x=3 odnosno 8.33 MHz pa eko opet neće na x=4 ili 6.25 MHz) što važi i za noviju generaciju ploča sa 80486. OPTI kolima i Award BIOS-om (i nude se isla mogućnosti) i novija ploča sa 80386 na 40 MHz. OPTI kolima i istim BIOS-ima.

Ako ne možete da utičete na brzinu sabirnice, moraćete da zamanite kartice onim bolim i skuplim, jedina alternativa vam je da ceo računar usporite. Što je besmisleno. Još jedna mogućnosi koju vredi ispitali se odnosi na broj balera pod DOS-om (1 baler = 512 B). Videli smo slučajeva kada je broj batera bio smanjen na svega 2 ili 3, iako DOS standardno odvaja 15 ukoliko nema drugačija komande u datoteci CONFIG.SYS. Pre nego što išle drugo učinite probajte da ubacite komandni red u CONFIG.SYS datoteku koja glasi: BUFFERS=20 zatim reselujte mašinu kako bi se nova komanda izvršila. Ponekade to pomogne bez ikakvog dopunskog petljanja a košta vas 10 240 bajtova DOS memorije.

OPTIMIZACIJA PROGRAMSKE BAZE

Prvi korak bi trebao da vam bude oslobađanje što većeg dela DOS memorije, nema većeg ubrzanja od onoga kede niz raznih pobudnih programa za miša, video kartu i druge do-



datke prebacite negde iznad granice od 640 KB. Tu se ponekada u zavisnosti od toga šta sve imate u računaru dobija i po 80 KB DOS memorije. Tipične situacije pobuda za mišе zauzme 5-10 KB, bafer (20) zauzmu 10 KB, pobuda za posebna režima video karta daljih 10-15 KB i najzad program za keš memoriju diska zauzme daljih 20-25 KB. Kada to saberete proizlazi da ste izgubili 45 60 KB dodajte tome još 20-25 KB za potrebe recimo mreže i eto vam bubitka od 65-85 KB konsne DOS memorije (uzgred upravo iz ovih razloga, test programi se koriste na računarima u kojima nema nikakve AUTOXEC.BAT datoteke, a CONFIG.SYS datoteka sadrži samo dva reda FILES=40 i BUFFERS=20 baš zato da bi se oslobodilo najveći mogući deo DOS memorije).

Ako imate računar sa procesorima 8088/8088 ili 80286, obavezno uzmite program pod nazivom ORAM. Ako imate fizički prisutnih bar 1 MB memorije (što su poslednji izdanci XT mašina imali, a stariji ne), kada ovaj program završi, ostaće vam na raspolaganju preko 600 KB slobodne DOS memorije čak i u najgoram opisanom slučaju. Slično drugim takvim programima koji rade samo na 32-bitnim platformama, i ova će jednostavno prebaciti (remapovati) sve pobudne programe iznad granice od 640 KB a ispod gornja granice od 1 MB, drugim rečima: iskoristiće inače neiskorišćen prostor u ovom segmentu. Za ves to će značiti ubrzanje od nekih 5-10% bez ikakvih dalјih zahvata (rezultat je dobijan merenjem brzine rada niza različitih programa).

Ako pak imate 32 bitnu platformu, onda izaberite jedan od dva najpoznatija programa ovog tipa na tržištu: birajte između *QEMM*-a i *386-to-the-Max* (o oba je bilo reči u sterijim brojevima). Nekakvo kumulativno iskustvo pokazuje da oba rade na veoma sličan način i da obe oslobađaju približno jednaku količinu memorije, *Max* ima malo bolju (automatizovaniju) proceduru instalacije (inače, u verzijama 5.1, oba rade sa *Windows 3.0* u ENHANCED režimu). Na instalaciju praktično ne možete uticati, ali u slučaju *Max*-a koga mi konstantno nemate ni potrebe, jer mu ništa ne može izbeći ako ikle posumnja, upteće vas šta da radi! Rezultat – radimo sa 603 KB slobodne DOS memorije.

Pre nego što pustite bilo koji od navedena dva programa, prvo instalirajte neki bolji program za keširanje tvrdog diska. Mi već standardno preporučujemo *PC Kwik Power Pak*, program koji ima najbolji disk keš i nekoliko drugih modula za ubrzanje video podsistema (posebno korstan za vlasnike Hercules kartica), tastature za prijam podataka za štampu (*Print spooler*) i za stvaranje RAM diska.

I ova program će se sam instalirati kao što rekosmo, u nekim prosečnim vrednostima bez obzira šte vam razna poruke na ekranu govore ime puno mesta daljoj optimizaciji. Neke kolege su imale problema sa ovim programom nema savršenog programa ali mi ipak varujemo da se radi o nepotpuno) instalaciji pre nego o nekoj grešci u programu. Mi ga imamo ne tri potpuno različite mašine iz različitih izvora i nikada do sada nismo imali problem se ovim programom.

Recimo, ova program će automatski upotrebiti bilo kakvu AT ili LIM memoriju na koju naiđe – i to sve na šta naiđe, lako je u principu tačno da keš treba da bude što veći, postoji niz situacija u kojima ne želita da vam beš sve ode na keš. Dalje posla DOS memorije druga najbrža memorije u računaru je AT (Extended) memorija, pa bi keš za najbolje rezultate trebalo smestiti u nju a ne u LIM memoriju. U slučaju programa *PC Kwik* to ćete postići ubacujući u komandni red datoteke POWERON.BAT iza reda koji pominje „superpck" komandu „/E XXXX", gde „xxxx" označava količinu memorije, a „E" da se radi od AT memoriji (E – Extended).

Razlike u performansama uopšte nisu zanemarive: za istu količinu memorije ovako odreden keš će raditi 5-10% brže nego kada koristite LIM memoriju a „xxxx" vam omogućava da zauzmete samo željeni deo raspoloživa memorija. Ostatak će vam OEMM ili *Max* pretvoriti u LIM memoriju pa teko možete raspolagati sa oba vrste u istoj mašini.

Druga interesantna mogućnost je da nerediте programu da kešira ne samo čitanje nego i pisanje, mada odmah napominjamo da to može biti veoma opasno ako vam nestane struja (sa njom nestaće i podaci koji nisu zapisani). Ako pošelite ovo da aktivirate iza komanda „E/ XXXX" napravite jednu pauzu i unesite komandu „/W+". U slučaju rada sa bazama podataka u kojima ima i dosta pisanje, ili bilo čime što zahteva i pisanje na disk ubrzanje je veoma veliko i kreće se od 50% do 700% (zavisi od brzine procesora, veličina keš memorije i brzine diska, navedeno važi za 486 na 25 MHz, 4 MB keša i disk Quantum 170A). Čak i na standardnim AT mašinama (286 na 16 MHz sa 1 MB keša), ubrzanje iznosi 50-250% što sa i te keko oseti. Ako imate UPS ne razmišlјa)te odmah to uključite.

Obavezno uključite poseban algoritam za keširanje („/O+") poseban algoritam za brz prenos sa i na disk („/H+"), a ako se opredelite i za keširanje pisanja uključite i proveru potrebe za pisanjem („/W+"), da ne gubite vreme ako ne trebe pisati. Pokušajte (eksperimentišite) i sa keširanjem celog sektora umesto samo jednog dela („/T+") a program će očitati disk da bi utvrdio koliko sektora po tragu imate.

Mi smo dobili dosta dobre rezultate sa svim ovim dodacima. Bez keširanja pisanja i u odnosu na varziju samoinstalacije popravili smo rezultate za prosečno 9% kada smo uključili i keširanje pisanja ukupne dobit je iznosila prosečno 19% (u operacijama sa pisanjem ubrzanja je iznosilo i do 420%). Mi konstimo 4 MB za keširanja diska u kontekstu brzine same platforme (486 na 25 MHz) praktično smo eliminisali tvrdi disk kao posebno usko grlo. Prema testovima „PC Magazina" verzija 5.6 ovako konfigurisan disk je prosečno 10.6 (8.3-12,4) puta brži od diska na računaru Compaq Deskpro 486, koji je poznat po svojoj brzini rada.

Namojte sa uopšta potresati što će vam recimo CORE test pokazati manju brzinu prenosa kada uključite keširanje pisanja na disk, samo pustite pomenute testove „PC Magazine"-a, deo sa varijabilnim veličinama datoteka i sva će vam biti jasno.

Tak kada ste ovako konfigurisali recimo *PC Kwik* (ili neki drugi program za keširanje), tek onda instalirajte *QEMM* ili Max. Pre toga rasplagaćete sa recimo 525 KB DOS memorije a kada ovi završe sa instalacijom i optimizacijom (koju morate tražiti odnosno potvrditi kao pošelјnu na dati upit) imaćete maltene sve pod kapom nebeskom i 603 KB raspoložive DOS memorije. Naravno pretpostavlјa se da raspolažete sa bar 4 MB RAM-e u mašini.

Najzad, efekti kombinacija ovakvih koraka optimizacije su zaista veliki. Naše probe ukazuju na ukupne prosečne razlike u brzini (mašinski i programski testovi) od 1.7.1 navše u zavisnosti od raznih elamenata (brzina računara, brzina tvrdog diska raspoloživa memorija itd). Čak i to je već veoma mnogo, ali trebe znati da taj prosek oduhvata i testova na čije rezultete ne možeta uticati bilo kakvom optimizacijom (recimo, poslovna grafika) ili ako utičete, taj uticaj je maltena nikakv drugim rečima tamo gde vam treba, lako se može desiti da ubrzanja bude daleko veće, recimo 4:1.

Sva ovo pokazuje da je veoma lako moguće da vi već sada bez ikakvih bilnih izmena posedujete računar koji faktički na konstite ke ko biste mogli pa makar se radilo i o 286 na 12 MHz, koga takođe koči pristup disku, spor odziv tastatura i slično. Proizilazi da lako možete ubrzati računar prostim potpunjim korišćenjem već postojećih resursa ili dodavanjem memorije. Sadašnja cena iznose oko DEM 85 po megabajtu (DIP/SIMM/SIP), ta cena može biti uzrok razmišlјanju ako imate ploču sa 286 na recimo 16 MHz koja sama košta oko DEM 165. No, ako imate recimo SX na 20 MHz ili 386DX na 25 MHz bez keša, najdelotvorniji način da sebi olakšate život i rad je da dokupite memorije i malo se pozabavite svojom mašinom. Satite se samo jedne stvari: na taktu od 25 MHz mašinski ciklus procesora 80386DX bez keša sa izražava u nanosekundama, e brzine diska još uvek u milisekundama već to je razlika od nekih stotinu hiljada puta.

Ko kaže da je vaš računar spor: kada procesor već sada ono ho radi u prazno dok ga periferije ne sustignu?

TVRDA ČETVORKA

Pre uzvesnog vremena, dosta detaljno smo pretresli ponudu poznate američke firme Quantum, čiji se proizvodi već poodavno prodsju i kod nas. Ovaj put smo pogledali šta nudi sigurno najpoznatiji nezavisni proizvođač tvrdih diskova na svetu, firma Seagate.

Dejan V. Veselinović

Seagate je proizveo do sada najodlja prodavan disk na svetu dobri stari model ST-225 svojevremeno nije bilo boljeg diska za modele PC i PC/XT. Radili su, radili, i radili i nikako da stanu. I dan-danas mnogi od njih sasvim lepo rade. Njihov najpoznatiji naslednik je bio model ST-251. U zavisnosti od toga da li je imao „-0" ili „-1" imao je već sasvim dobru brzinu od 40 odnosno 28 mS, no već su sa njim počeli izvesni problemi. Taj model je već umeo da priređuje neprijatna iznenađenja, a pri tom na mislimo na YU ludilo povezivanja ovog MFM diska na RLL kontrolere, kako bi se isterala veća brzina pranosa i 50% veći kapacitet.

No firma Seagate je pala u nemilost sa svojim modelima 137/237 koji su već imali prečnik od 3.5 (umesto odiadašnjih 5.25) inča jer su po svemu sudeći imali ugrađenu grešku na ležaju motora koji je vrlo brzo lanjire. To je dovodilo do pojave da disk jednostavno odbija da se zavrti. Sa tim modelom jedno vreme dosta popularnim kod nas, Seagate je došao na loš glas. Oni su sa svoje strani radili sva i svašta, ali koliko znamo, bez većeg uspeha. Zatim su pretprošlu i polovinu prošle godine provali kupujući razna firma i ograničile firmi čime su izbili na treće mesto u svetu po proizvodnji diskova (iza IBM-a i firma Fujitsu).

Onaj relativno mali dodir sa njihovim diskovima najnovija generacije nam je pokazao da se po svemu sudeći Seagate dobro trgao. Novi modeli dobre performanse, agresivne cene – sve to bi trebalo da budu sastojci formule za uspeh na tržištu. Pa ipak, ostala su nam dva pitanja na koje nismo imali odgovor: može li Seagate da povrati izgubljeno poverenja kupaca, i kako se ti diskovi porede sa konkurencijom, što ljudom to bolja?

Da bismo to utvrdili, na sto smo stavili ukupno četiri modela firma Seagate, u rasponu od 42 do 210 MB, pored njih smo stavili i tri konkurenta drugih poznatih imena firmi Quantum, Connar i Fujitsu – i dali se na posao.

Svi diskovi su testirani na jednoj platformi koja koristi procesor i486 na 25 MHz sa sekundarnom keš memorijom od 128 KB. Namerno smo se opredelili za ovu platformu, jer smo želeli da u što je moguća većoj meri eliminišemo računar kao izvor kašnjenja. U svim slučajevima koristili smo DOS 3.3, bez ikakvih dopunskih programa za keširanje i slično. Jedina dva reda u CONFIG.SYS datoteci su bila „files=40" i „buffers=20". Svi diskovi su prilikom testiranja na prvoj particiji sadržali tačno 11 MB podataka.

Ambijentni uslovi su namerno bili podešeni u okvire idealnog: konstantna temperatura od 24 stepena, vlažnost vazduha 65%, a verujemo da visina (VIII sprat) nije uticala na rezultata. Ukupno testiranje je trajalo 67 časova, tokom sedam dana.

Quantum LPS52A

O ovom disku smo već dosta govorili, pa ćemo ga samo ukratko razmrsiti. Visok je 2.54 cm, pračnika 3.5 inča i formatiranog kapaciteta od 52 MB. Glavna vrlina mu je veoma mala potrošnja struja, manje od polovina nekog proseka (6 VA), kao i veoma mali stepen zagrevanja. Druga velika vrlina mu je brzina, koja čak i u ovoj klasi predstavlja u najmanju ruku ozbiljnu konkurenciju i mnogo većim i skupljim diskovima.

Prosečna brzina prenosa iznosi 898.6 KB/s, vreme slučajnog pristupa je 17.7 mS, vrama čitanja susednih tragova je 4.21 mS, a vrama pristupa disku kroz DOS 23.01 mS.

Najzad, da završimo sa ovim diskom jednom prostom konstatacijom – ovaj disk je naša referenca u ekonomskoj klasi, koja je ujedno i apsolutna referenca.

Quantum 170A i Fujitsu M2615E

I o ovim diskovima smo već dosta govorili. Quantum disk je za sada naša referenca u klasi velikih diskova, a nasladio je Fujitsu disk. Quantum ima odlične performanse (mada ga je konkurencija već sustigla), a Fujitsu igra na kartu pouzdanosti i čak izuzetno povoljne cene, u konačnici žrtvuje apsolutne performanse. Prvi ima formatiran kapacitet od 168 MB, a drugi 180 MB.

Seagate ST357/AX

Ovaj model treba da zameni stariji model sa oznakom ST157A. U osnovi radi se o istom disku, varovatno sa nekim unutrašnjim poboljšanjima. Prečnik diska je 3.5 inča i veoma je lagan, dosta lakši od konkurenta, modela LPS52A. Pločica sa elektronikom je puna komponenata sa obe strane, a kola su na žalost okrenuta ka unutrašnjosti diska, pa nismo mogli da ih razgledamo. Ima 820 cilindara, 6 glava i 17 sektora po tragu.

Brzina direktnog pristupa se pokazala kao za današnje pojmove sporih 24.5 mS – i tu je sve stalo. Naime, ma koliko se trudili, ovaj disk je jednostavno odbijao da radi na našoj testnoj mašini. Zamenili smo kontrolor, ali bez ikakve sreće. Što je najlepše, mi smo ga pripremili za rad (FDISK i FORMAT) bez ikakvih problema. Zapisao je dve diskete programa i podataka, a kode bez ikakvih problema, ali kada je došlo do čitanja, jednostavno je zaglavio.

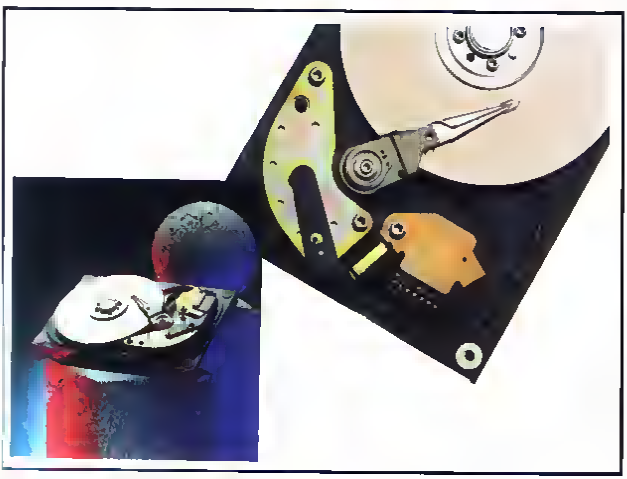
Pomislili smo da smo bili „srećka" koje su dobile onaj jedan primerak u „n" hiljada koji položi sve testove ispravnosti i uprkos tome ipak ispusti dušu. Stupili smo u vezu sa prodavcem od koga smo ga dobili, na šta su oni zapanjili i brže-bolje zamenili disk drugim koji su sami formatirali odnosno probali. Rezultat je bio potpuno isti – lepo piše, neće da čita.

Ovo do sada nismo imali prilika da vidimo. Naši standardni vezniei su jedan anonimni tajvanski proizvod i drugi (takođe tajvanski) proizvod sa oznakom „Maxtor". Nikada nam se nije dogodilo da nam bilo koji disk odbije poslušnost na bilo kom od ta dva kontrolera, a posebno ne na ovako selektivan način.

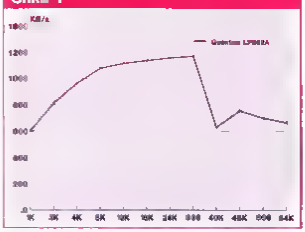
Očigledno nismo mogli izmeriti performanse, a tu je i ovaj nemio događaj. Mi ćemo se zainteresovati i probati još nekoliko takvih diskova, a u međuvremenu – nemojte kupovati disk koji je toliko probirljiv prema vezniku, sem ako ga ne dobijate u okviru cele konfiguracije (tada je rad problem prodavca). Bar za sada.

Conner CP3000

Formatiran ovaj disk ima kapacitet od 42 MB, poraklom je od firme koja je verovatno najce-

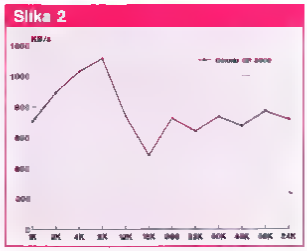


Slika 1



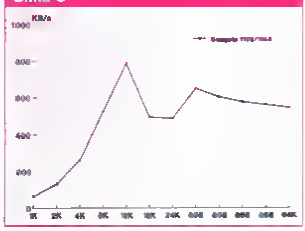
Brzine prenosa za Quantum LPS52A

Slika 2



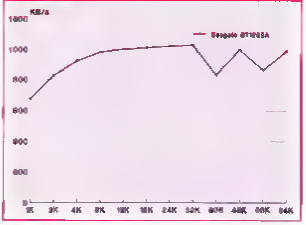
Brzine prenosa za Conner CP3000

Slika 3



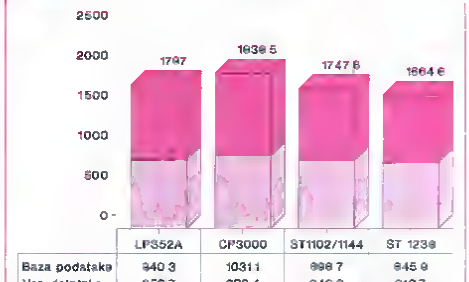
Brzine prenosa za Seagate ST1102/1144

Slika 4



Brzine prenosa za Seagate ST1239A

Slika 5



njenja u svetu danas za male diskove ili bar ima „imidž" Rols-rojs klase I zaista kada ga vidite odmah vam je jasno da se radi o proizvodu na koji je utrošen veliki rad Finalna obrada je izvanredna disk ja ih da ližI ne može biti, a troši neverovatnih 4 25 VA što objašnjave odsustvo bilo čega što bi se moglo nazvati „povšenom temperaturom" Jednostavno nakon što sate rada van kućišta odnosno na prirodnoj ventilaciji disk ostaje doslovno hladan na dodir

Ima prečnik od 3,5 inča xxx cilindera x glava i xx sektora po tragu Elektronika je najblaže rečeno uzorno sređena u celosti konsti SMT tehnologiju a pločicom dominira VLSI čip firme Cirrus Logic sa oznakom „CL-SH260 15C C" Pored njega tu je jedva još 15 čipova od čega su dva memorijska kola

Disk smo naravno formatirali bez ikakvih problema a naravno i radio je bez ikakvih problema sa oba pomenute vezrika (kažemo „naravno' jer se to od firme Connar i očekuje – bez ikakvih problema) Prosečna brzina prenosa je bila 768 2 KB/s, što je nekada bilo fantastično, a sada je već ispod proseka (slika 2) Slučajan pristup je iznosio 26 20 mS, a redni pristup 7 64 mS Pristup disku kroz DOS je iznosio 28 42 mS, to je za današnje pojmove dosta sporo mada nije previše strašno za ovu klasu Najzad ukupno vreme rada na svim testovima je bilo najgore

Ovaj disk se sada povlači iz prodaje jer ga zamenjuju noviji i odlji modeli Mi smo ga probali zarad onih koji će se opredeliti za ovaj disk kako zbog imena tako i zbog verovatno osetno sniženih cena usled želje prodavaca da se otresu lagera slanih modela Ako se bavite poslovima koji ne zavise mnogo od diska i ako ga nađete po povoljnoj ceni slobodno ga kupite

Seagate ST1102/1144

Iako se radi o različitim kapacitetima ova dva diska su po mnogome svemu nalik jedan drugome kao jaje jajetu Oba imaju prečnik od 3 5 inča, oba su u dobro obrađenim kućištima oba imaju dobre performanse i veoma povoljne cene Takoda napominjamo da oba čak i ne oko izgledaju daleko ozbiljnije i solidnije od svog manjeg i mlađeg brata sa kojim nismo imali sreće

Oba su sasvim lepo radila sa oba pomenuta vezrika Jedine nedoumice oko ovih diskova je propratni papir na koma se nalaze sve specifikacije potrebne sa rad all je broj sektora navođen u vidu tri brojke To je zaista čudan običaj a radi se da bi se ukazalo prodavcima da se broj sektora može menjati u navedenom rasponu U principu opredelite sa broj od 33 ili 34 jer ako se držite ostalih podataka sa 35 već više neće redlti (bar ne pouzdano)

Da su oba modela izvučena iz zajedničke platforme se vidi i po merenjima performansi koja se poklapaju na nivou +/- 1 5% u apsolutno svim testovima Zato smo se opredelili prvega iz razloga prostora da ih prikažemo zajedno Slika 3 prikazuje brzinu prenosa koja baš nije pohvalna medutim to ovim diskovima nije ni najmanje smetalo da zabeleže čak veoma dobre rezultate Posebno hvalimo vreme čitanja susednih tragova 1 04 mS za 1102 i 0 99 mS za 1144 neverovetni rezultati u oba slučaja

Diskovi su veoma tihi, a stepen zagrajavanja nakon više časova neprekidnog rada je zadovoljavajući (toplo ali ne i vrelo) Koliko znamo cene ovih diskova su i više nego povoljne za kapacitete koje nude, deluju ozbiljno brzi su pa ih smatramo dobrom kupovinom

Podsecamo da je jedan od njih model 1102 referenca za poslovnu klasu mašine (R/Optima SX) posle 6 meseci rada još uvek odličan

Seagate ST1239A

Ovaj disk već spada u tešku kategoriju Ima prečnik od 3 5 inča ali i kapacitet od 211 MB nakon formatiranja Parametri koje velja znati su 818 cilindara 14 glava i 36 sektora po tragu Za razliku od manje braće ovaj disk spada u grupu „Swift modela a ima indikacija da je zaista zemišljen tako da se uspešno nosi sa konkurencijom u ovoj već veoma ozbiljnoj klasi

Slika 4 prikazuje „krivu" brzine prenosa. Oveko dober oblik smo videli još jedino kod Conner diska od 210 MB bez obzira na malo kolebanje od 32 KB nagore Vreme slučajnog pristupa je 15 49 mS a vreme čitanja susednih tragova 3 92 mS Pristup disku kroz DOS je rekordnih 17 68 mS a odalle na dalje sve je samo rekord do rekorda (mada su razlike katkada čisto akademske prirode) Ipak ovo je do sada najbrži disk sa kojim smo se susreli i to u programskim testovima koje smatramo jedino merodavnim za sve Prosečna brzina prenosa iznosi 930 9 KB/s

U slučaju programskih testova ovaj disk je bio 29 9% brži od Fujistu diska i daleko skromnijih 3 1% brži od modela Quantum 170A Ovaj drugi ga prevazilazi po krajnoj obradi i solidnosti (masivnosti) oklopa kao i po nešto manjem stepenu zagrejavanja To je naša jedina zamerka ovom disku dosta se greje mada nije strašno neprosto je nešto topliji od recimo Quantuma

Ovo je po svemu sudeći odličan disk ozbiljan konkurent u visokoj klasi Deluje verovatno dosljno tih je i veoma brz jos osnovne da mu se potrefi dobra cena Ako ga nađete za dobru cenu kupite ga najverovatnije nećete pogrešiti

Korisna adresa

Preduzeće MOST
11000 Beograd Kralja Bodina 2
Tel 011/422 170 Fax 011/426-144

AKO KUPUJETE VGA

Prošlo je dosta vremena od kada smo se pozabavili video prikazima; u međuvremenu, dogodilo se dosta toga – od pojave novih ili novijih standarda, do zaista korisnih tehnoloških novina, koje su gotovo odmah uvedene u prodaju. Sa druge strane, jasno je da među nama ima dosta onih kojima dobra grafika i te kako treba za uspešnije bavljenje poslovima. Zato se ponovo vraćamo ovoj temi.

Dejan V. Vasiljinović

Klasična VGA grafika je maltene (tehnološki) zaboravljena, mada naravno živi kao standard. Od kada je IBM prikazao novi grafički standard pod inventivnim imenom XGA o staroj VGA više niko ni ne govori. Pregledi proizvođača monitora pokazuju da su i oni manje više digli ruke, pa je lako novi standerd postao tzv. Super VGA (SVGA). Radi se o rezoluciji od 600×600 sa nominalno 16 boja, a realno u velikoj većini slučajeva i sa celih 256 boja odnosno nijansi. Naravno, bez preplitanja.

VRSTE KARTI

Iste te kartice najčešće mogu da postignu za još jedan korak veću rezoluciju od 1024×768 ali sa preplitanjem. Sa 512 KB memorije ove rezolucije može istovremeno konsisti samo 16 boja iz palete od 16,7 miliona ukupno mogućih boja. Učestanosti sinhronizacije je najčešće 35,5 ili 36 KHz što odgovara maksimumu od 600×800 bez preplitanja.

Preplitanje u ovom skučaju znači naizmenično osvežavanje svake druge linije na ekranu, a navodi se kao važan podatak jer od učestanosti osvežavanja zavisi stabilnost slike. Što je ta učestanost veća to je slika stabilnija i jasnija, manje je treperenja i manje je raznih egzotičnih efekata po ekranu (do kojih inače dolazi u slučaju jeftinijih i lošijih monitora). Računar se može ubrzati ugradnjom krišaša na veću učestanosti, dok se video karte time ne ubrzavaju ali im raste rezolucija. Naravno, da bi neka rezolucija mogla i u praksi da se koristi i monitor kao i karta moraju biti sposobni da rade na datoj učestanosti, kako bi se mogli sinhronizovati.

Postoji dosta dilema oko učestanosti osvežavanja linija na monitoru, odnosno oko postavljanja nekog minimuma. Problem se svodi na to da postoje dva režima rada, sa i bez preplitanja, ako se radi u režimu bez preplitanja, kada video karta redno osvežave svaku narednu liniju bez preskakanja, slika će biti stabilne već i na učestanosti osvežavanja od 50 Hz, ali ako radi sa preplitanjem, da bi slika bila stabilna biće potrebna osetno veća učestanost osvežavanje linije. Iskustva sa IBM 8514/A standerdom i učestanosti osvežavanja od 43,5 Hz jasno pokazuju da to nije dovoljno, pa postaje mnogo jasnije zašto se insistira na mnogo većim učestanostima.

Kao neko opšte pravilo, možete smatrati da učestanosi osvežavanja u režimume bez preplitenja nije problem ako je broj veći ili jednak 55 Hz, a za rad sa preplitenjem, sve iznad 65 Hz je sasvim dobro (i za 49,4% više od 8514/A).

VIDEO PROCESORI

Danas na tržištu postoji čitav niz video procesora, jer se svaki video čip može smatrati procesorom. Od popularnih modela poznate su firme Oak Technologies (Oak), Western Digital Imaging (Paradise), Trident Technologies (Trident), Cirrus Logic i Tseng Labs. Redosled nije ni malo slučajan, jer ovako kako smo ih naveli idu i cene video kartica na bazi ih procesora. Pored navedenih postoji i niz drugih firmi koje imaju slične proizvode na osnovu neke sopstvene tehnologije kao recimo Chips & Technologies, ATI itd.

Posebno interesantni su C&T koji je nedavno prikazao poseban paket od dva kolo namenjen emulaciji standarda 8514. To ne samo na programskom nego i na nivou registara (dakle, totalna kompatibilnost), ali naravno sa nižom prednosti pre svega većom brzinom rada. ATI je poznata firma iz Kanade koja se odavno proslavile svojom serijom „Wonder" („Čudo") video kartica, ali i modemima, i oni su prikazali novi model ATI 6514/Ultra karticu koja daleko prevazilazi IBM 8514/A po brzini rada može da radi sa ili bez preplitanja sa 256 boja, čak 8514/Ultra" nema VGA grafiku na sebi, postoji drugi model, pod nazivom „8514/Ultra+VGA" koji ima na sebi sve što i mladi brat, naravno sa standardnom VGA grafikom, pa čak i „bus" veznikom za miša, koji se isporučuje uz karticu.

Posebnu kategoriju čine dva video procesora poznate firme Texas Instruments 34010 i 34020. Pošto se radi o veoma moćnim procesorima, postoji mogućnosi da se oni „naviju" na čak veoma velike radne takove, čime se dobija na brzini. Tako model 34010 lipično radi na 50 MHz, a ove čipove i na 60 MHz, dok se inherentno jači model 34020 (32/32 bita) za sada zadržava na skromnijih 30 ili 32 MHz.



Da bi ovi procesori mogli da se uklope u postojeće standarde, potrebna im je posebna programska veza sa postojećim programima, poznata pod skraćenicom TIGA (Texas Instruments Graphics Adepter) do pre godinu dana ovaj veznik je bio prava retkost, ali danas ga već ima ne sve strane. I to u standardnom izdanju programe (recimo čak i popularni program „DrawPerfect" namenjen najširim masama, standardno sadrži ovaj veznik). Naravno, ma šta drugo proizvođač kertice podržavao ili ne, dva veznika mora da ima: onaj za *Auto CAD* i onaj za *Windows 3.0*.

PROBLEM BOJA

Proizvođači video karti najčešće insistiraju na rezoluciji, jer se to do sada najlakše postizalo. Činjenica je da rezolucija predstavlja veoma važan elemenal vernosti slike, ali je isto tako činjenica da se sa po tom pitanju već osnovni VGA standard veoma približio običnom televizoru, a svi znamo da ti kućni aparati mogu imati odlične slike. Rezolucija od 600×800 o većima da i ne govorimo, je već podoste iznad rezolucije TV prijemnika, zašto onda slika na ekranu ne deluje ni izbliže oneko dobro kao na TV-u?

Odgovor ima dva dela. Prvi se odnosi na „reckavost" slike na monitoru odnosno na relativno grube korske od linije do linije. Taj problem se naziva „aliasing". U osnovi on nastaje leko šlo se svaki piksel (najmanja adresabilna jedinice prikaza) trelira sa svega dva stanja – ili je osvetljen ili nije. Dakle, da bisle dobili prelaz naredni piksel iza željenog se ugasi a onaj odmah ispod njega pomeren nadole za jednu liniju, se upali, naše oko ovo percipira kao korak ili slepenik. Ako bismo međulim isli taj pa čak i susedni piksel mogli samo delimično da osvelimo odnosno zatamnimo, korak bi za naše oko poslao manji, ne bi se leko jasno video. Sedašnji trendovi u računarskoj industriji su lekvi da se se ovim problemom proizvođači sve bolje i bolje hvataju u košlac. Zapravo, video čip na određeni način ugrađen u logiku hardvere, izmeše boje u prednjem i zadnjem planu lako da prevan naše oko kome se učini da je prelaz daleko ravnomerniji, pa samim tim i verniji. Takvi čipovi sa ugrađenom „anti-aliasing" logikom, se danas već nalaze na tržištu (Sierra CEG/DAC čip).

Drugi problem se odnosi na najveći broj boja, a u apsolutnoj paleti i u broju koji se može prikazati odjednom. Tu leži osnovni razlog kvalitata TV slike, pošlo se radi o analognoj tehnici, broj boja koje se odjednom mogu prikazati je praktično beskonačan. Na žalosl, video kerte još uvek teže ka tom cilju, mada ima veoma interesantnih rešenja za sasvim razumne pare.

Standardne VGA grafike konsli 4-bitnu boju, to znači da istovremeno može prikazati samo 16 boja. Pošlo je celokupna paleta boja definisana sa 24 bita, apsolutan broj raspoloživih boja iznosi 16,7 miliona. Bolje kertice današnjice sasvim dobro rade sa 8-bitnom definicijom boje, odnosno sa 256 istovremenih boja. Idealno, kada bi radile sa svih 24 bita, broj istovremenih boja bi bio jednak apsolutnom broju raspoloživih boja, odnosno 16,7 miliona. Takve slike su već na nivou najbolje moguće fotografije. Na žalosl, takve karte su po ceni na nivou veoma dobrih automobila, relke su i tražje posebne pobude itd. Ukratko, nisu mnogo praktične.

Srednje rešenje je pronađeno u 15-bitnom definisanju boja (odnosno 16-bitnom) koje omogućava istovremen prikaz 32 766 (odnosno 65 536) boja iz palete od 16,7 miliona. Iako ovo već zadire u zonu tzv. „duboke boje" i uprkos lome da je lo za ogromnu većinu nas i više nego dovoljno. Ipak nije apsolutno na



nivou kvaliteta fotografija mada se deformacije primećuju tek kod izuzetnih fotografija. Ovo rešenje je veoma zgodno i za proizvođače i za kupce i zbog svoje zaista umerene cene: veće za svega 10-15% u odnosu na standardna kartice današnjice.

Usko grlo u prikazivanju odja je DAC, odnosno digitalno u analogni konvertor. radi se o čipu (najčešće od Inmos-a i Brooktree-ja) koji ima zadatak da digitalni signal pretvori u analogni i kao lakav ga isporuči monitoru. Ima naravno i drugih problema, ali neki od savremenih video čipova, kao recimo Tseng ET4000, već imaju ugrađenu podršku za 15i odnosno 16-bitno definisanje boja. Dve firme, Edsun Laboratories i Sierra Semiconductor, danas prodaju posebne DAC čipove koji rade sa 15i odnosno 16-bitnom definicijom odja. U celosti su kompatibilni do nivoa nožica sa postojećim DAC čipovima na tržištu, pa se njihova ugradnja svodi na prostu zamenu jednih drugima.

Pre nego što otrčite da ih nekako, ko zna kako, nabavite i ugradite, treba da znate još dve činjenice. Prvo, da bi ti čipovi mogli da rade kako treba, BIOS na video kartici mora da utvrdi (pomoću POST rutina prilikom podizanja sistema i kartice) o kakvom se DAC-u radi, drugim rečima, pored ugradnje novog DAC-a, možda (vrlo verovatno) ćete morati da zamenite i BIOS na kartici. Drugo, sami programi moraju da imaju sposobnost prikazivanja većeg broja boja. Tehnički, nije nikakav poseban problem doraditi programe tako da mogu, ali većina jednostavno nema taj dodatak, pošto se radi o sasvim novim tehnologijama koje nisu postojale kada se pisao osnovni kod. Dakle, najbolje će biti da zamenite celu karticu novom koja će biti prilagođena novoj tehnologiji.

Povećavanja broja odja će sigurno bitno povećati vernost slike, ali zbog većeg broja proračuna, to će bez sumnje usporiti rad video podsistema. Ovom problemu, pored ugradnja veoma brzih procesora, proizvođači pristupaju na više načina.

Prvi i osnovni je zamena jeftinih RAM čipova skupljim VRAM čipovima (VRAM – Video RAM), ovim se omogućava dvostrani (dual-ported) istovremeni pristup memoriji od strane video procesora i od strane centralnog procesora (CPU), naravno, na različitim adresama. Rezultati ovog rešenja su veoma varijabilni – bilo je modela koji su pokazali veoma solidna ubrzanja, odk su drugi radili isavim isto kao i sa standardnom memorijom.

Drugo, daleko ređe ali i daleko skuplje rešenje je da sa na video karticu stave dve vrste memorije. I to u pozamašnim količinama. Prva vrsta je VRAM, koja služi isključivo za rad video procesora, a druga je običan DRAM koji služi za nizanje naredbi i podataka koja tek treba obraditi. Prva vrsta omogućava izuzetno brz rad video procesoru jer je i brza i potpuno posvećena njemu, bez ikakvih preklapanja i preplitanja, a druga omogućava veoma brz prijem i dalje prosleđivanje podataka, čime se centralni procesor veoma brzo oslobađa za dalji rad (podatci se prebacuju u RAM na kartici). Nije ništa neobično v delj savremena kartice sa recimo 512 KB VRAM-a i 1 MB RAM-a, često prošireno na duplo više u oba slučaja, a ako sadrže i posaban odaljak za VGA grafiku, imaće i posebnu memoriju za nju.

Drugo rešenje je, kao što rekosmo odje ali treba imati u vidu i to da će ono bar udvostručiti cenu kartice. I ne zaodravilo, veća rezolucija ne znači više detalja na istoj površini dela slike, nego više slike na istom ekranu.

PROBLEM MONITORA

Nije ništa neobično da se sva pažnje posveti video kartici, a da se monitor koji treba da finalizuje celu operaciju, sasvim izostavi iz razmatranja. Prećesto se događa da gro para ode na karticu, a na monitor šta preostane. Naravno da je ovo sasvim pogrešna filozofija, ako ni zbog čega drugoga, onda bar zbog slabe poslavka da najslabija karika određuje snagu celog lanca.

Razlog zbog koga je to lako je sigurno ce na. odbar monitor košta i veoma dobre pare, često 2-10 puta više no sama kartica. Tolika razlika zavisi od kvalitete i veličine (po dijagonali) same katodna cevi i naravno kvalitete elektronike koja sloji iza nje. Nije nepoznat efekal da se ista katodna cev koristi u dva monitora sa veoma različitim cenama i naravno veoma različitim kvalitetom slika, dakle, ne odbacujte olako kvalitat elektronike.

Pošto svi savremeni moniton u boji (bar mi ne znamo za izuzetak) koriste analogne ulaze broj odja koje mogu da prikažu je neograničen. U tom kontekstu, nemale problema od najeftinijeg do najskupljeg, bez razlike. Međutim, u slučaju najveće radna razolucija postoje velika razlike. Ogromna većina današnjih monitora bez ikakvih problema radi do razolucija od 800×600 bez i do 1024×768 sa preplitanjem odbar deo, za ipak poprilično veće pare (20-40% više) može de izgure 1024×768 i bez preplitanje. Ovi drugi najčešće imaju i veću dimenziju po dijagonali (16 umesto 14 inča) a sve češće se pojavljuju za sada veoma skupi modeli monitora sa dijagonalnim merama od 17, 19 pa i 21 inč (21 inč = 53 3 cm).

Ako video kartu birate po rezoluciji i brzini rada, monitor birajte po finoći lačke vernosti odja dijagonali i stepenu (odsustva) zračenja. Finoća lačke se izražava u delovima milimetra. Iipične vrednosti su 0 31 i 0 26 mm, mada ih ima i se finocom od 0 25 i 0 24 mm (nekiko su veoma skupi). Kao stvar opšteg principa, gledajte da finoća na bude veca od 0 28 mm. To danes više nije neka retkosti niti neki poseban luksuz.

Za svakodnevnu upotrebu dijagonala od 14 inča (35 6 cm) je sasvim dovoljna i predstavlja neki razumen kompromis između cene i kvalitata za lola profesionalniji rad u CAD/CAM oblasti ili stonom izdavaštvu, bez 16 inča (40 6 cm) nemojte ni počinjati sam ukoliko vam je budžel izuzetno nalegnut. Samo ce oni najbogatiji razmatrali modela sa ekranima od 19 (48 2 cm) i 21 (53 3 cm) inč, pre svega zbog cena.

CENE, TRENDOVI I MODELI

Najzad, dolazimo do konačnog razmatranja šta kupiti.

U ekonomskoj grupi od video kartica pogledajte one sa Tseng ET4000 video čipom provereno su dosta brže (10-25%) od najjućeg konkurenta firma Trident, mada su i nešto skuplje. Pošto se radi o generičkom proizvodu (isti proizvođač mnogo različitih imena) lako je dali neko posebno ime primere radi, u našoj verziji se prodaje kao Morse KP1024 TVGA a mi smo je videli pod bar još 7 8 imena). Tipična minhanska cena je oko DEM 270-310. Kod nas je prodaju Gama Electronics, MlacoM i drugi.

Alternativno pogledajte (ako imate prodor do američkog Iržišta) ATi model „8514/Ultra+VGA", stvama cena je oko USD 500 što jeste dosta skuplje, ali je to koliko znamo najbrže klasične (tj. bez TI 304XX) video kartica često brža i od onih sa TI procesorima.

Profesionalna grupa bi trebala da se zainteresuje za dva proizvoda. Prvi je Hercules Graphics Station GB1024-2, ova kartica koristi procesor TI 30410 na 60 MHz, ali na sebi sadrži i celokupnu VGA grafiku, osnovni model ima 1 MB VRAM-a, a jači bral porad ovoga ima i 2 MB DRAM-a. Oba modela nude punu 24-bitnu graliku (16 7 miliona odja odjednom) u rezoluciji od 512×480 (kao TV slika!), verovatno nameno odabranoj zbog sličnosti sa nekim Targa video kartama. U SAD karte koštaju oko USD 800 i USD 1 200.

Ako vam treba brzina bez ikakvih uslova, razmislite o jednom novom modelu američke firme National Design Inc. pod imenom „Volante AT1000". Kartica ima odvojenu VGA grafiku (Tseng ET4000 video čip sa 512 KB proširivo do 1 5 MB) sa posebnim odeljkom za

TIGA veznik. Umesto uobičajenog TI 34010 kartica koristi jači procesor TI 34020 i to na zdravih 32 MHz. Standardno se isporučuje sa 512 KB brzim baker memorije i 1 MB (proširivo do 4 MB) VRAM-a. Sem toga ima i podršku za TI 34082 numerički koprocesor za slučaj da se zovete Han Solo i da sla svraćali iz Zvezdanih ratova onako naviknuti na brze od sve ilosti. Nedostatak ove kartice je ograničenja na 256 boja i najveća razolucija od „samo" 1024×768. Cena je dosta razumna za ono što dobijate: oko USD 995.

Od monitora u ekonomskoj klasi bismo izdvojili dva modela. Prvi je Philips monitor pod zvucnim nazivom „Brilliance" radi se o monitoru sa dijagonalom od 14 inča i zaista lepom slikom vernim bojama malim ugaonim izobličenjima i malim zračenjem (u saglasnosti sa najnovijim švedskim standardima). Radi do 600×600 bez i do 1024×768 sa preplitanjem. Možete se raspitati za njega kod Game Electronics u Beogradu.

Drugi model sa kojim smo veoma zadovoljni (bar za ovih mesec dana rada) je monitor firme MITAC model AL-4020. Praktično je potpuno isti po svemu sa opisanim Philips-om sem u slucaju cene: po kojoj je osetno jeftiniji. Njega prodaju firme Mlakar & Co u Austriji i MlacoM u Ljubljani. Ovaj monitor, uzgred budi recano, je jedini do sada koji nas je naveo da posumnjamo na bateriju u instrumentu za merenja elektro-staličkih zračenja. Iek kada smo malaci fizički pnslonili uz ekran uprkos svim pravilima rada sa lim uređajem igla se jedva primetno pomerila. Sa cenom ispod 20 000 dinara ovo mora da buda dobar posao.

Od boljih i nužno skupljih monitora nama se dopao Sony Multiscan HG" (u Evropi ima našlo drugačiju oznaku). Pomalo je cudan jer nominalno ekran ima dijagonalu od 13 inča ali taj kvalitet slike a pre svega boju. Treba videti da bi se varovalo. Veličina tačke je 0.28 mm a cev je naravno tipa „Trinitron". Evropsku varziju ovog monitora prodaje beogradska firma Mosi.

Ko ima para i potrebe pa mu treba svih 16 inča na mora da se dvoumi. EISO model 9070S (u S A D se zove „Nanao 9070S") je najbolje sto smo do sada videli, ne samo u ovoj klasi. Veličina tačke je 0.26 mm a kvalitet slike je izvanredan, ništa manja. Na žalost isto vazi i za cenu: oko DEM 2 000 u Nemačkoj. Kod nas prodaje ko stigne a kupuje ko može.

Od velikih monitora koji se nama bar na dopadaju mnogo navodimo da mnogi proizvodaci nude bar po neki model u ovoj egzolicnoj klasi. Tu ipme kod nas koje isporučuju velike monitore (recimo: IDEK Hitachi, Mitsubishi, NEC) su Gama Electronics, Most i MlacoM/Mlakar & Co. savetujemo vam da se dobro raspitate a za ta para i pogledate ako ikako možete sva modala koji vas zanimaju jer oni ceslo kostaju kao bolje opremljen računar.

Korisne adrese

Pošlo adrese pomenutih domaćih dobavljača možete videti iz oglasnog dela li sla navodimo samo adrese nostranih izvora:

National Design Inc., 9171 Capital of Texas Hwy., Houston Building, No. 230, Austin, Texas 78759, S A D., tel. 991-512 343 5055

Hercules Computer Technology Inc., 921 Parker St., Berkeley, CA 94710, S.A.D., tel. 991-415 540-6000

ATI Technologies Inc., 3761 Victoria Park Avenue, Scarborough, Ontario, Canada, M1W 3S2, tel. 991-416 756 0718, fax 991-416 756- 0720

NOVI VGA ŠPIL

Tehnika ide napred, a sa njom se menja i struktura tržišta – sa Hercules grafike se polako ali sigurno sve više prelazi na VGA video podsisteme. Uz opšti pogled na tržište video kartica u tekstu „Ako kupujete video", pripremili smo i test četiri konkretna proizvoda.

Dejan V. Veselinović

U svetu video čipova nema baš previše proizvođača a ono što ima jasno je podeljeno u razna kategorija kvaliteta i cene. Na dnu po ceni se nalazi firma „Oak Technologies", sledi je firma „Western Digital", „Chips & Technologies" i „Trident", a na vrhu piramide se nalazi firma „Tseng Labs" (ili kako to Tajvanci kažu „Ceng"). Prema našim dosadasnjim iskustvima kao najbolje su se pokazale karte sa Tseng ET4000 video čipom.

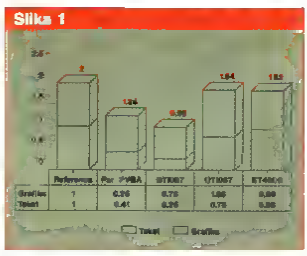
PARADISE PVGA1A

No naravno, ni drugi na spavaju: naprotiv, rade iz sve snage i dalje se trudeći da budu što bolji i što jeftiniji. Recimo: kartice sa Oak video čipom kakve smo jednom imali u ruci pre godinu dana su već ova pula zamenjene novijim modelima. Firma Western Digital Imaging koja je vlasnik firme Paradise je upravo izbacila u prodaju svoj novi video čip WD 9901C sa jasnom namerom da popravi svoju tržišnu poziciju a isto vazi i za firmu C&T. Ma kako da bismo voleli da probamo je nove modele nismo uspeli da ih se dokopamo, a sa druga strane glavni krijenjum nam je bio ono što se kod nas nudi odnosno što možete sada kupiti ovde ili u Minhenu.

Kao platformu za testiranje smo upotrebili našu referentnu mašinu u profesionalnoj klasi sa procesorom i486 na 25 MHz. Za ovo smo se opredelili iz tri razloga: prvo želeli smo da što je moguće više odstranimo uticaj brzine rada računara, drugo: pretpostavlja se da će za ovakve karte najzainteresovaniji biti profesionalci nakonjani CAD/CAM-u i sličnim delatnostima, a oni se njima neće baviti sa AT mašinama na 16 MHz i treće: empirijski smo videli da će odnosi brzine ostati manje više isti na veoma jakim i relativno slabim platformama.

Uslovi testiranja su standardni: kontrolisana ambijentlan temperatura od 24 stepeni, vlaznosl vazduha 65%, te standarni testovi koje i inače koristimo, mada u dvostruko vecem obimu radnih modela (kako zbog većeg maltretiranja samih karti tako i zbog brzine platforme).

Slika 1



Na brzinakom testu

Duplikalna video memorija na računaru je bila uključena (pošto postoji neravzeno od video karti) a sem nje nismo koristili bilo kakav drugi program (čisti DOS). Referentna video karta je MORSE KP1024 TVGA (Gama Electronics Beograd) a referentna mašina u profesionalnoj klasi je opisana u „Računarima" 75, str. 8-9 (MlacoM/Mlakar & Co.).

NIKO NE SPAVA

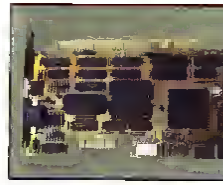
Ovaj model se praktično nalazi pred gašenjem jer se očekuje zamena novijim i razradenijim video čipom (što bi trebalo da mu jako obori cenu). Kartica iako je dobra pokazuje znake slarosli kao i nekih dosta čudnih efekata.

Ima standardnu dvotrećinsku dužinu, uobičajenu za slanje proizvode loga tipa. Oslanja se na Paradise video čip sa oznakom PVGA1A, koji je podrzan od strane 13 TTL čipova, 2 PAL čipa, 18 memorijskih čipova (VIA V53C464AP80L, 80 nS) monliranih na posto ja, 2 ROM čipa, 1 DAC čipa i 4 kristala (25.17/26.32/36/42 MHz). Jedina je karta u ovoj grupi koja poseduje veznik za prolaz (feature connector"). ovo je pomalo začudujuće s obzirom na dosta veliki broj karti koje se nadovezuju na VGA kartice ali izgleda da je trend da se taj veznik izbaci iz upotrebe.

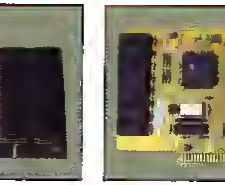
Veoma solidno je napravljena sa dobrim dalovima. Priručnik nije los, mada smo videli i mnogo bolje ilustracije su dosta primilivne, svakako ispod nivoa samog proizvoda. Programski veznici koje bi trebalo da dobijete uz nju (mi nismo) su ili prilično bajati ili potpuno prevaziđeni, što je nesumnjivo minus. Kao izvesna kontrateža ostaje činjen ca da dosta veliki broj savremenih programa ima vec ugradenu podrsku za bas Paradise karte, pa vam veznici ponekada nisu ni potrebni.

U testiranju ova karta je odbila da radi u dva video režima: režimu 7 (80×25 mono) i režimu 15 (640×350 EGA mono). Ispitali smo ovo i ustanovili da ova karta po dalinciji neće izvršavali mono režime na kolor monitorima a hoće na mono monitorima. Dakle iako je neuobičajeno ovo ipak nije stvarna mana, jer da se ne uplašila ako probate a ona neće. Više zabrinjava njen pad na testu statusa registara jer bi jo moglo da znaci da će biti situacija u kojim programi neće umeti da se snađu u radu sa njenim registrima.

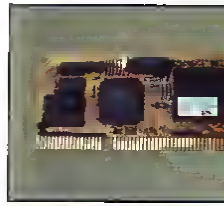
Karta inace ima niz bubica: sve do VGA režima radi sasvim normalno. Može da posluži na razoluciju od 800×600 sa 16 boja, ali ništa preko toga: no uprkos lome, naš test program je pokazao da sam video cip podrzava rad do 1024×768, a da se proizvodač, ko zna zašto opredelio da taj režim ne ponudi. Takođe ako želite rad sa 256 boja, moracete da se ograničite na rezoluciju od 640×400 (!) dakle nivo čak ispod standardne VGA rezolucije. Čudno zaista.



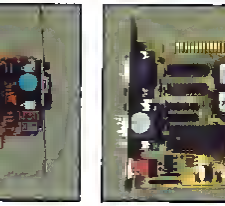
Paradise PVGA1A



Tseng Labs 4000



Oak OTI037C



Oak OTI067-AO6

Nenad Marjanović

Drugi aspekt po kome se vida njene godine je učestanost osvežavanja ekrana najmanja u grupi (62 3 Hz u lekalualnom rezimu) mada je u VGA rezimu radila na nivou svih ostalih Ovo nije ozbiljna mana all ipak šlo više lo bolje U konlekstu brzine, zauzela je drugo mesto od pozadi mada su razlike relalivno male

Sve u svemu jedan ozbiljan proizvod sa nekim manama koje odaju njen vek Pošto će biti zamenjena verovatno će biti nuđena za male pare, pa ako nemate izbora uzmite je ali ipak provenle da li možete da se dokopale dru gih boljih modela drugih proizvođača.

OAK OTI037C

Kartica spada u red najeftinih na Iržištu a nema nikakvu prepoznaljivu oznaku sem vi deo čipa Nema ni neku dokumentaciju (bar ova koju smo mi dobili) ali ima jednu disketu sa vezničima i konsnićkim programima Prilič no je jednostavna poludužinska je a sadrži Oak Technologies OTI037C video čip UMC" DAC dva BIOS EPROM-a četiri TTL kola osam RAM čipova (u SMT lehnologiji TI TMS 4464-12, 120 nS) i četiri kristala (25 175/26 32/ 35 5/40 MHz) Jedina je u ovoj grupi sa dva priključka za TTL i analogne monitre Odlično ja napravljana

Polozila je mirne duše sve leslove kompatibilnosti sem već uobicajenog pada na laslu interapta vertikalnog brisanja (Vertical blank interrupl), rak rane ogromna većine video karti U kontekstu brzine rada zauzela je ubedljivo poslednje mesto uglavnom zbog veoma dugog vremena potrebnog za naposredna pristup ekranu

Radi se o jednom (opet) verovatno zastarelom proizvodu koji zbog raspoložive memonje ne može da radi na rezoluciјama vecim od 800×600 Uz karticu cale dobili veznik za Win dows 3 0 šlo je dobro ali sam rad je dosta kilav

Ovu karticu gledajte da zaobiđete i nabavi le novije verzije uprkos veoma povoljnoj cani

OAK OTI067-AO6

Karticu cela veoma lako prepoznati – ubed ljivo je najmanja video karta koju smo do sada videli Veoma lici na najnoviju ganaraciju AT Bus veznika i I/O karti lajvanskog porekla zau zima lačno onoliku dubinu kolika je potrebna za 16-bilni vaznik a po visini je nešto veća od polovine standardne kartice Na njoj se nalazi najnoviji video čip firme Oak pod nazivom OTI067 cetiri memonjska čipa (dva na postoljima, dva na pločici u SMT lehnologiji HY51C4256S-10 100 nS) jedan DAC jedan BIOS EPROM dva kristala i jedan poseban Oak čip, sa oznakom OTI069 Njegov zadatak je da pomoću dva kristala (14 318/50 MHz) ga nenše sve potrebne učestanosti sinhronizacije

Ova kartica se isporučuje sa neobično velikom bibliotekom veznih programa što je još lepše, radi se o veoma aktuelnim vezničima, mada ih ima i za stanje verzije. Recimo kartica podrzava *AutoCAD 11* pored naravno obavezног *Windows 3.0* Sa priloženih 512 KB me morije može da radi do razoluciје od 1024×768 u 16 boja sa prepltanjem odnosno 800×600 u 256 boja bez prepltanja Kao veo ma nov proizvod uspela je da zbuni program za ispitivanje osvežavanja slike po kome ona radi samo do VGA rezoluciје ali sa najboljim vremenom osvežavanja (70 12 Hz do VGA i 59 96 Hz za VGA) Polozila je bez ikakvih problema sve leslove kompatibilnosti sem narav no onoga za vertikalno brisanje

U konlekslu brzine ova kartica se pokazala kao druga najbrža, svega 3% sporija od naše reference Medutim, do ovoga je došlo isključi vo zbog veće brzine referenca u lekstualnom rezimu (za 24%) pošto je u graličkom rezimu ova kartica bila 8% brža od reference Drugim rečima, ako se bavile isključivo grafičkim apli kacijama smatrejte ovu karticu brzom od refe rence

Sa cenom od oko DEM 160 (sa 512 KB) ne mozete promašiti sa njom, mala je jeftina ima odlicnu podrsku i brza je kao davo

TSENG LABS 4000

Radi se o genenckom proizvodu kartici ko ja sa pravi na jednom mestu a zatim prodaje pod mnogo imena Oslanja se na Tsang Labs ET4000 video cip jednim DAC čipom 5 TTL kola i čak In BIOS EPROM-a (dve standardna i trecim sa oznakom „VGA-Sync ROM") Na kartici se nalaze dva kristala (40/? MHz) a po mocu NS DM74LS245N čipa i posebnog ROM-a se dobijaju sve ostale potrebne učesta nosti Memonjski čipovu su američke proizvod nje (MT 4C4256-8 80 nS) u ukupnom iznosu od 1 MB

Ova kartica je već treća generacija proizvo da na bazi ovog video čipa U proeđenju sa našom ralerencom koja je prva generacija ova je kudikamo praznija (manje delova) sa još nekim razlikama Oko 11% je sporija od re ference (-7% u lekstualnom i -13% u grafičkom režimu) efekal o koma smo vec govorili za koji smatramo da poliče od generalora učesta nosti umesto klasičnih oscilalora za svaku uce stanosl posebno Sa druge strane ona je jedi na kartica do sada koja je polozila sve leslove kompatibilnosti uključujuci i notorni lesi varti kalnog zamračenja

Druga razlika se odnosi na učestanosl osvežavanja ekrana u režimu rada od 800×600 gde je ova kartica radila ne sa 55 66 Hz kao relerancna vec sa 59 65 Hz 4 Hz na izgleda mnogo ali se na našem monitoru lo ipak videlo u nekim programima kao osetno manje vidljiv broj linija u pozadini najzad u principu gledano ova učestanosl treba da bu de šlo veca U režimu rada od 1024×768 sa prepltanjem obe karta su dale jednak rezultat 86 94 Hz (ko tvrdi da mu slika trapari ili ima los monitor ili oči Terminatora)

Praklično ceo ovaj lesl se sveo na poređe nja ove kartice i nove Oak kartice sa novim video čipom Oak kartica je osetno brza (+24%) u graličkom režimu i sporija (-22%) u lekstualnom rezimu pa bi joj po lom kriterju mu dali prednosl sa druge strana Tseng kartica radi na 1024×768 sa vecim brojem boja (256 umesto 16) sa ali i bez prepltanja šlo je čini bezuslovno lehnicki savršenijom i pogodni jom za rad u CAD/CAM-u ulisak koji je donekle pokvaren pomalo bajatim veznim programi ma (AutoCAD do V 10)

Bajzad jednačina cena ova kartica košta oko DEM 270-320 ili 68-100% više od Oak kartice Ako imala monitor sposoban da radi na 1024×768 bez prepltanja dileme nema uzmila Tsang karticu ako imala monitor koji na 1024×768 radi samo sa prepltanjem i ako vam baš treba 256 boja, uzmite Tseng karticu a u svim drugim slučajevima dobro razmislite

* * *

Za sve zainlaressovane – pošlo se nejveci deo video karti prodaje pod jako mnogo imena kao jedina garancija slicnosti sa opisanim mo dalima su oznake na video cipovima Zbog to ga smo ih i navodili u lekslu a sem toga pez ljivo pogledajla i priložene lotografije

Kao šlo vidimo lehnika je dobro napredo vala. Tseng koji je do sada vladao manje vise suvereno više nije sam na vrhu drugi su se približili a mnogi se približavaju Prijatno smo iznenađeni novom Oak OTI067 karticom zbog sada vec odlicnog kvalitala uz i dalje izuzetno povoljnu cenu Ovo je, mislimo rezultat rada proizvođača koji zaista zali da uspe od počet no dosta sumnjivih proizvoda vidimo da je do gurao do veoma ozbiljnog konkurenta čak naj boljem na trzislu

Ovakvom filozofijom dobijamo pre svega mi konsnici jer nam se za relalivno male pare nudi odlican proizvod Pobednik u ovoj grupi bar moralni je Oak OTI067 Ita li ce da urade u narednoj itaraciji?

Kontakt adrese

Oak OTI037C Tseng ET4000
Preduzece MOST
YU 11000 Beograd Kralja Bodina 2
Tel (011) 422 170 Fax (011) 426-144
Oak OTI67C Tseng ET4000
MlacoM
YU-66100 Ljubljana Celovška 185
Tel (061) 193-010 553-276
Fax (061) 193-016

PUNI MODELI

Jedno od glevnih unepređenja *AutoCAD 11* u odnosu ne steru verziju je *AME* (Advenced Modelling Extension) – puni (eolid) modeler tele. Ieko se potpuna AME verzije doblje tek uz dopletu ne osnovni peket, ogromne mogućnosti koje donosi solid modelirenje čine da se 90% kupeca *AutoCAD 11* odluči i ze AME. Šte se podrezumeve pod pojmom „puno modeliranje". I koliko je ono korleno?

Nebojša Novaković

Kao što ste u drugim tekstovima mogli uočiti postoje tri osnovne vrsle prostornog (3-D) modelirenja na računarnme (razlikujmo pojam modelirenje od pojme prezenteclje modele) To eu žičeno površineko i puno (solid) modelirenje Žičeno modelirenje definiše ivice koje obrazuju konlui u objekte Model objekta je tada providan i vide se jedino vektori ivica objekte – model tade neliči na žičani kavez – otude i engleski naziv „wirelleme modelling" Se ovekvim modelirenjem nlje moguće nikakvo senčenje jer jednostavno nema površine koje bi bile senčene Rezime 3-D telo kao skup 1-D vektore

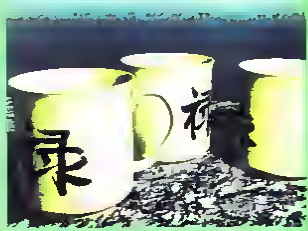
Površinsko modeliranje (*surface modelling*) definiše objekat kao skup površine koje mogu imati i svoju teksturu i meterijalizaciju radi foloreallstlcnog senčenja Površinsko modelirenje omogućave i generisanje složenih površine kao teljafe, vinskih češe ili kupole ruskih crkava Složene površi se tade mogu dobljati rotaciJom krivih oko osa rotacije ili klasičnim načinima dobijenje prevoizvodnih poviši Rezime 3-D telo kao skup 2-D površina

SAVRŠENO OBLIKOVANO TELO

Puno modeliranje (*solid modeling*) je najsavršenlji vid oblikovanje tela na računeru Solid model tela sadrži Informacije ne samo o ivicame i površinema već i o unutrešnjosti tele To podrazumeve podatke o gustini, čvistoći i drugim lizičkim karakteristikama Takvo potpuno sagledavenje tele omogućave analizu i simulaciju njegovih svojstave, kao ne primer otpornosti ne naprezanja ili statike, samo uz pai dodatnih programskih redove

Ono što je, međutim, najbitnije i najlepše u svemu ovome je mogućnosi logičkih operacija (unlja, presek razlika) ne punim modelima Šte one donose? Zamislimo na primei da treba napraviti model kocke sa kosom cilindričnom rupom u sredini U solid modeleru jednostavno definišemo kocku i kosi cilindar u prostornom koordinetnom sistemu preko dve odgovarajuće naredbe Ova dve elementa su kao i lopte pnzme, piramide rotacione tele itd primarni elementi – „solid primitives" ne kojima vršimo željene operacije U ovom slučaju vršimo logičku operaciju razlika kocke i cilindre, i kao rezullet operacije dobljamo novo telo – kocku sa cilindričnom rupom u sredini Svaki solid modeler će vam onda bez probleme izračunati (za zadanu vrstu meterijala) i masu površinu, zapreminu, centei mase i niz drugih parametara Ne ovom telu možemo potom vršiti i operacije zaobljavenja ivica, preseka po nekoj ravni Ili pak vršiti neku drugu logičku operaciju sa trećim telom čiji neki deo zapreme isti deo prostora kao i prelhodno telo Novodobijeno telo će opet imati i nove karakteristike.

Veoma bitne stvar je da se prilikom ovekvog formirenje i oblikovanje modela čuva sve informecije o telima čijim „kombinovenjem" je model nastao Te informecija se čuve u posebnom logičkom CSG stablu (*Constructive Solid Geometry tree*) u kojem sa može od početka do kraje pietiti stvaranje modela. U skladu s tim se i eventualne propusti ne gotovom modelu napravljenl u nekom od prathodnih koraka njegovog stvaranje mogu lako i brzo kongovoti Ako smo, na primer, u pravljenju modele od makli do sedmog korake a ne patom koraku je jedna od solid primitive bile suviše široka, mi jednostavno možemo (u AME naiedbom Change Primitives) promeniti dimenziju (u ovom slučaju suziti) tu solid primitivu, a sam program će u llenutku ponovo prelačunati korake koji slede do poslednjeg i prepraviti model

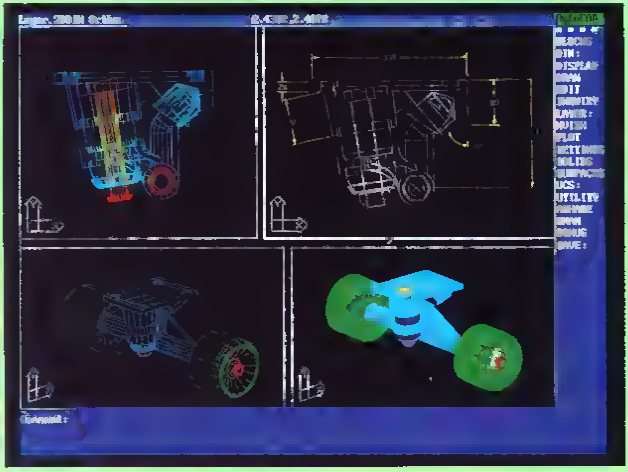


Puni model jepenekih šolja

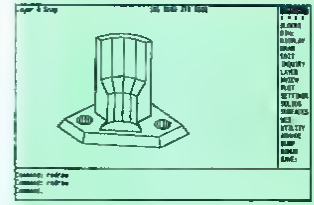
AUTOSOLID (1989-1990)

Puno modelirenje je zbog svojih veoma velikih herdverskih i softverskih zahteve bilo potpuno nepristupačno svetu personalnih računara sve od 1989 godine UNIX grafičke radne stenice koje su do tada bile jedina platforme za ovu vletu modeliranja te su godine izgubile monopol Naime puno modeliranje u pravom emislu reči se te godine prvi put pojavilo kod više PC CAD pakela u različitim cenovnim razredima Dva primsie su karaktenčna *DesignCAD 3-D* od verzije 2 nadalje u klasi jeftinih, i *AutoSolid* u klesi skupljih paketa Prvi je po prvi put donео svetu (ne baš potpun) solid modeler za manje od 300 USD u ono doba a drugi, *AutoSolid* je bio prvi specijalizovani strogo solid modeler za PC *AutoSolid* je bio prvi Autodeskov paket u 386 kodu sa ugrađenim *DOS Extenderom* i prvi se konsnićkom okolinom nalik *UNIX X Windows* Vrlo ozbijno i pametno zamišljen AutoSolid nje poživeo ni dve godine Razlog konsnički interfejs pravashodno Ljude u *Autodesku* je iznenadilo to da je najveći broj potencijelnih (e i stvemih) korisnika *AutoSolid* pakete bio iz grupe korisnika *AutoCAD* Bez obzira na eve kvalitete konsničkog interfejsa *AutoSolid*-a, *AutoCAD*-ovci su naučili da rade onako kako su naučili i novi korisnički interfeje im je smetao

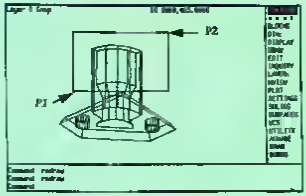
Još jedan razlog je i što se tokom 1990 pojavilo više veoma ozbiljnih PC CAD paketa sa ugrađenim solid modelirenjem, kao *Anvil 5000* i *Silverscreen* u klasi cene AutoCAD-a.



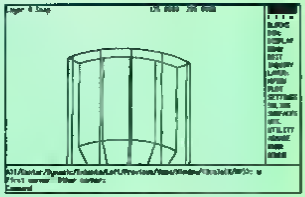
***AutoCAD 11 AME* puni model točkove ekejtborda**



Primer punog modela



Zoom prozor



Uvećani prikaz zakivljenih ivica

Autodeskov glavni izvor novca i slava ja na taj način postao ozbiljno ugrožen. Uz brojne sugestije za solid modeler u AutoCAD-u, donela je odluka da se solid modeler *AutoSolid*-a malo unapredi i integrše u *AutoCAD 11* pod nazivom AME. Sam *AutoSolid* je time prestao da postoji.

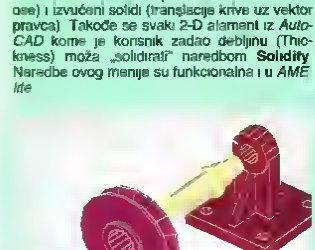
POGLED NA AME

U dve kutije u kojima se *AutoCAD 11* paket dobija, *AME* čine poslednje dve od 8 instalacionih disketa, *AME* priručnik i list papira na kojem je odštampan kod za konšćenje *AME* ne odgovarajućem serijskom broju *AutoCAD*. Prilikom prve instalacije i startovanja naredbe „load AME" iz Solids menija program vas pita za taj kod i bez njega na možete koristiti *AME*.

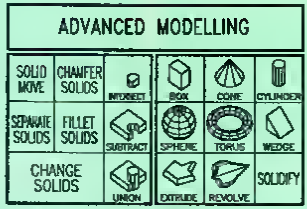
Oni koji ne doplate *AME* na raspolaganju imaju *AMElite* osakaćenu verziju *AME* sa generisanjem najosnovnijih primitiva bez logičkih operacija. *AMElite* ja tu ponajviše zato da bi sa u takve *AutoCAD* bez problema mogli unositi solid modeli izrađeni sa *AME* na drugom sistemu.

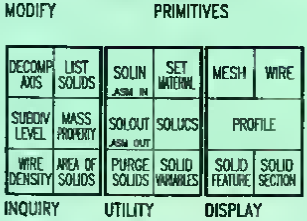
Dokumentacija koja prati *AutoCAD* spada u red onih koje su za svaku pohvalu. Pored *AutoCAD 11 Reference Manuala* i manjih knjiga posvećenih *AutoLISP* programskom jeziku, *ADS AutoCAD C* razvojnom sistemu (u kojem je i *AME* napisan) instalaciji kao i priručnika za početnike, i *AME Reference Manual* vrlo dobro obavlja svoj posao.

Kada je *AME* učitan umesalo menija **Solida** pojavljuje se **Solid Primitives** (Sol-Prims) meni, osnovni meni *AME*, u kojem su naredbe za generisanje solid primitiva. Pored neposrednog generisanja osnovnih geometrijskih tela (kvadar, kupa, veljak, lopta, torus, prizme) mogu se dobiti i obrtni solidi (kriva rotirana oko neke ose) i izvučeni solidi (transjacije krive uz vektor pravca). Takođe se svaki 2-D element iz *AutoCAD* kome je konsnik zadao debljinu (Thickness) može „solidirati" naredbom **Solidify**. Naredbe ovog menija su funkcionalna i u *AME lite*.



Puni model bez skrivenih linija





Alatke za oblikovanje punih modela

Znatno zanimljiviji je **Solid Modify** meni. Ovaj meni sadrži logičke operacije preseka unije i razlike dva solida (**Intersect, Union, Substract**) zasecanje i zaobljavanje ivica tela (**Chamfer i Filler**) pomeranje i obrtanje po željenoj osi željenog koordinatnog sistema (**Move**) promena parametara solid primitiva koji čine model (**Chenge Primitives**) i razdvajanje solida na primitive koji ga čine (**Separate**). Pomoću ovih naredbi možemo od osnovnih činilaca – solid primitiva stvoriti praktično bilo kakvo telo. Ove, kao i sve ostale naredbe funkcionalne su samo u *AME*.

Solid Inquiry meni sadrži naredbe za pregled i podešavanje fizičkih karakteristika solid modela i podešavanje različitih sistemskih promenljivih u raznim mernim sistemima. Naredbe **List Solid** prikazuje CSG logičko stablo solida ili koordinate izabrane ivice ili površine. **Mass Property** naredba izračunava i prikazuje fizičke osobine jednog ili više izabranih solida, kao masu, zapreminu, koordinate kvadra u koji je objekt upisan, centar mase i proizvod i momente inercije. Ovi podaci se naravno mogu snimiti i kao poseban tekst fajl. **Solid Area** kao što samo ime kaže, izračunava ukupnu površinu solida. Tu su takođe i naredbe za računanje karakteristika u imperijalnom (stopa i funta) CGS (centimetar i gram) ili SI (metar i kilogram) mernom sistemu.

Solid Display meni sadrži više naredbi vezanih za prikaz punih modela. Mesh i Wireframe naredbe određuju da li će solid biti prikazan kao puno telo sa spoljnim površinama ili samo kao „žičani kavez" ivica. **Copy Feature** naredba omogućava kopiranje izabranih ivica i površina solida, što je dosta korisno za tehničke crteže delova tela. **Cut Section** generiše presek kroz telo koji se, kao poseban entitet, može potom kroz **Paper Space** *AutoCAD*-a naposredno ubaciti u tehnički crtež. **Profile** naredba iscrtava projekciju tela na željenu projekcijsku ravan sa skrivenim linijama prikazanim ili ne. Projekcija, kao poseban entitet takođe može kroz **Paper Space** „postati" tehnički crtež. U ovom meniju su još i naredbe za podešavanje promenljivih koje definišu preciznost izračunavanja i gustinu (broj ivica ili površina) pokaza zakrivljenih tela.

Poslednji, **Solid Utility** meni sadrži naredbe za izbor i definisanje materijala od kojih će puni model biti „izraden", podešavanja ostalih sistemskih promenljivih, podešavanje konsničkog koordinetnog sisteme sa ivicom ili površinom tela. U ovom meniju su takođe i naredbe za unos ili slanje modela iz/u *AutoSolid* privremeno brisanje podataka vezanih za solid iz RAM da bi se rad ubrzao i kada smo se naužívali solid modeliranja, brisanje *AME* ili *AMElite* iz RAM.

SPORO ALI NEZAMENLJIVO

Obično se smatra da se puni modelen, pa time i *AME* u *AutoCAD 11* jedino koriste u mešinstvu i delimično u građevinarstvu. Zasta, modeliranje elemenata u mašinstvu ispitivanje i simulacija njihovih mehaničkih karakteristika i time znatno brže proizvodnja prototipa nezamislivi su bez solid modelera.

Međutim, čak i sam Autodesk je prevideo koliko su velike koristi od punog modeliranja i u arhitektun „Probijanje" vrata i prozora u zidovima, preseci kroz zgradu i projekcije modela tako neophodne za arhitektonske tehničke crteže ili konstruktivna analiza objekta su samo neki od primera.

AME jeste relativno spor i svi ovi podaci o modelu gutaju jako veliki prostor u DWG fajlu (a i u RAM). S druge strane sve naredbe *AME* su neposredno dostupne konsniku kao C rutine u ADS razvojnom sistemu tako da se mogu prošinvati i podešavati po želji. Na taj način relativno lako se mogu stvarati i sopstvene specijalizovane aplikacija sa *AutoCAD 11* + *AME* kao osnovom.

Sve u svemu, cena *AME* dodatka je oko jedna šestina cene osnovnog *AutoCAD 11* paketa, ali se taj dodatni izdatak u svakom slučaju isplati.

Korisna adresa

OSA računarski inženjering
11130 Beograd, Mice Stojkovića 8
Tel 011/557-592, Fax 559-845

OD PODRUMA DO KROVA

AutoCAD AEC Architectural je modul koji radi u _AutoCAD_ grafičkom okruženju, namenjen posebno arhitektonskom projektovanju. Kao i slični programi koji se oslanjaju na _AutoCAD_, pisan je u _AutoLISP_-u, i, što predstavlja malo iznenađenje, proizvod je Autodeska. Za manje upućene, čuveni _Autoshade_ i _Autoflix_, kao i veliki broj aplikacija i modula, razvili su nezavisni autori, kasnije ih je otkupio Autodesk.

Đorđe Grujić

AEC je skup LISP rutina i simbola koji se koriste u arhitektonskom projektovanju. Vrlo dobro razrađena rutina omogućavaju lako istovremeno crtanje i prostornog modela objekta i svih potrebnih osnova. Potpuna i vrlo obimna biblioteka simbola obuhvata praktično sve neophodno — od sirujnog prekidača držača za peškire u kupatilu ili sušača za ruke do sportskih terena i staza za atletiku, vozila i parkinga.

SA LISPA NA C

Rodan novembra 1987. AEC je prohodao u dobro poznatom okruženju _AutoCAD_ verzije 9. Kao i sva ostala aplikacije, pratio je razvojni put osnovnog programa i najnovija verzija 2.11 koristi kao podlogu _AutoCAD V11_. Još jedna značajna izmena vezana i za izmenu osnovnog programa je poslepeni prelazak sa _AutoLISP_ a na C programski jezik.

Jedna ni malo nevažna osobina AEC-a je da program sam prati na kom LAYER-u i UCS-u se radi. Ako potrebam LAYER ne poslopi AEC ga sivara sa prethodno određenim imenom bojom i vrstom linija. Ova na izgled samo zgodna osobina koja korisnika oslobađa stalne provere gde se u crtežu nalazi dobija na težini kada se zna da je struktura LAYER-a koja je ugrađena u AEC za _AutoCAD 386 V11_ prihvaćena kao British Standard, s preporukom da bude i standard EEZ od 1992. Mislim da ne treba posebno naglašavati šta ovo znači. Eto bar jednog polja rada gde smo već na nivou Evrope!

Ovom prilikom se nećemo posebno zadržavati na instalaciji programa, kao i na pratećoj dokumentaciji. Uslov za instalaciju je već instaliran _AutoCAD_ odgovarajuće verzije i serijskog broja, 8 MB slobodnog prostora na disku i 6 MB RAM memorije. Preporučujemo isle konfiguracija računara kao i za _AutoCAD_ (vidi avgustovska „Računara"). Program će raditi već sa 2 MB, ali je takav rad toliko mukotrpan da ga nikome ne preporučujemo. Naglašavamo. Ja se AEC vezuje za serijski broj instaliranog _AutoCAD_-a — sve druge kombinacije izazivaju velike napnilke! Konfiguracija instaliranog _AutoCAD_ a važi i za AEC. Dokumentacija se sastoji od knjižice A4 sa logički raspoređenim (po radosledu menija) opisom komandi. Ovo nikako ne znači da je i nedovoljna. Naprotiv, sve je opisano detaljno, sa lačnim oznakama gde se šta nalazi na grafičkoj tabli, odnosno u kom meniju, ako radite s mišem. Utisak je da uputstvo pretpostavlja iskusnijeg korisnika koga neće zbuniti često pozivanje na _AutoCAD_. Ne zaboravimo da je, sve vreme dok sle u AEC-u, vaš dobri stari pouzdani _AutoCAD_ sa kojim ste proveli toliko besanih noći, tu i da ga možete svakog trenutka pozvati.

LAKOĆA PROJEKTOVANJA

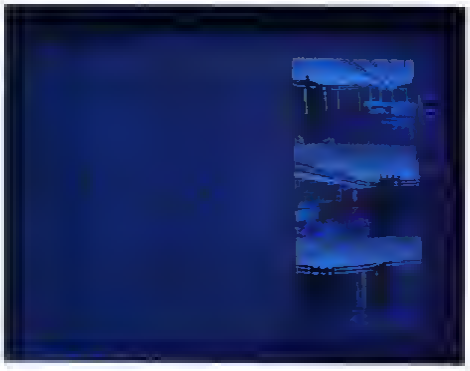
To može u početku zadavati male probleme, pogotovo onima koji su navikli da komande kucaju. Naime, samo _AutoCAD_ komande se kucaju, dok AEC komande morate birati sa menija ili grafičke table. Mora se paziti i prilikom prekidanja izvršenja komandi. Prekinuta AEC komanda može da izazove čudne posledice ako je u trenutku prekida vrednost brojnih promenljivih koje program prati izmenjena.

Rad započinja biranjem formata papira i razmere u kojoj ćete crtati. Za razliku od _AutoCAD_-a gde korisnik sam odredi koju će mu stvarnu veličinu predstavljati jedna jedinica, AEC pretpostavlja da je to 1 mm. Malo nezgodno za arhitekte koji su navikli na izražavanje dimenzija u centimetrima, ali ja to jedna od pojedinosti koje pokazuju da programer ipak nije arhitekta po obrazovanju.

Zatim podešavale parametre objekata u crtežu. Pod time se podrazumevaju spratna visine, visine parapeta, slojevi fasadnih i pregradnih zidova, vrste detalja spoja prozora odnosno vrata sa zidovima i drugo. Naravno da su sva ova veličine promenljive i u kasnijem toku rada. U program su već ugrađena vrednosti po BS, tako da nećete imati nikakvih problema prilikom izrade projekata za inostranog investitora.

Jedan od važnijih parametara crteža je da li će se raditi u ravni ili prostoru. Program omogućava rad u ravni koji zahteva manji utrošak vremena i memorije, do samog kraja kada je jednom komandom moguća sve osnove načiniane u ravni preobražiti u prostorni model objekta.

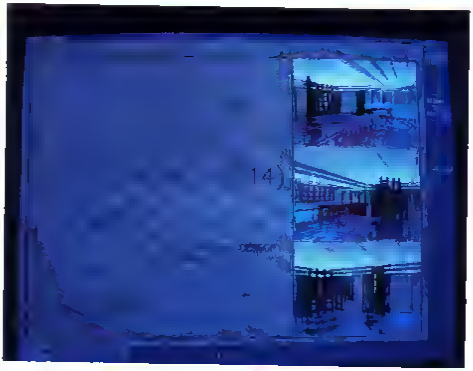
Najnovija verzija AEC-a podržava izradu modela objekata do 99 spratova. Počinje se od nultog, odnosno prizemlja. Organizacija prostora se može izvršiti na tri načina: skidanjem prostornog dijagrama, odnosno oblika prostorija, dimenzionisanih tako da kontura prolazi osom obodnog zida, crtanjem zidova, ili kombinovano. U prvom slučaju AEC će završen dijagram pretvoriti u osnovu komandom SPACE TO WALL, uz uslov da mu označite spoljnu konturu objekta. U drugom preseke zidova cele morali sami da očistite, upotrebom komande CLEAN UP INTERSECTION, koja ima podop-



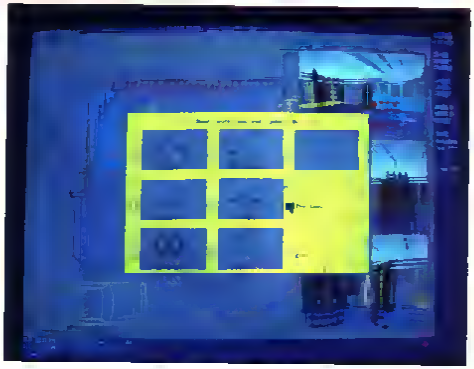
Prostorni model enterijera — osnova i tri perspektive generisane komandom PERCEIVE



Prozor za dijalog sa vrstama slepenica



Izbor iz biblioteke simbola unutrašnje opreme

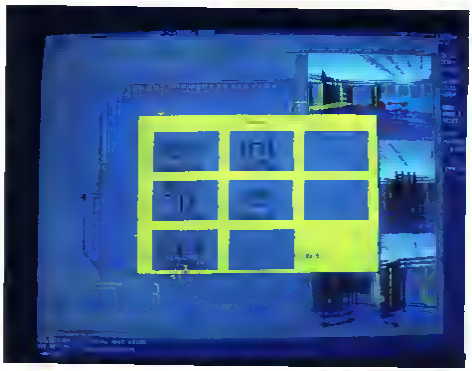


Izbor iz biblioteka simbola aportaklih tarene

cije za sva moguće slučajeve spoja dva zida

Kada ste nacrtali zidova sledi postavljanje otvora Prozore i vrata možete birati iz biblioteke simbola ili pravili svoje ako ne nađete nijedan koji vam se svida AEC će sam napraviti otvor uzidati vrata ili prozor i omalterisati keko fasadu Jako i unutrašnjosi Mali nedostatak ove operacije je što se mora ponoviti za svaki prozor odnosno vrata - komande *AutoCAD-a* — za kopiranja i umnožavanje ne daju zaljeni rezultat

U ovom trenutku imate osnovu sa otvorima Za izvođački projekat je to dobra podloga ali investitora i buduće kupce više zanima raspored nameštaja i enterijersko uređenje uopšte Nema problema na raspolaganju je bogat izbor nameštaja i upotrebnih predmeta kojime se može odrediti čak i vrsta završne obrade Postoji i bogata biblioteka simbola biljaka tako da za dva sala rada može da se lako uradi na primer uređenje enterijera prodavnice ili poslovnog prostora



Prozor za dijalog sa vrstama vrata

Sve što ste nacrtali na jednom spratu možete jednom naredbom iskopirati na proizvoljan broj viših ili nižih spratova Ovo možete uraditi još u početnoj fazi rada, ili na samom kraju pod uslovom da su svi spratovi (ili većina) isti ili slični kao na primer kod poslovnih zgrada U nacrtanu osnovu možete dalje umetati konstrukciju instalacije oznake i opise po želji

Na prvi pogled AEC je najkonsniji za poslove kao što je ovaj upravo opisan — manji objekat za koji treba pribaviti saglasnosi investitora bez ulaska u tehničke detalje Međutim, isline je drugacija Svaki simbol je blok sa promenljivim nivoom detaljnosti ainbula To znači da jedne vrata mogu biti prikazana samo kao simbol ali i sa potpunim tehničkim opisom od dimenzija do vrste šarki proizvođača i boja odnosno završne obrade Još lepše program sve atribule pamti i slaže u tabele žaljenog formata (DBASE LOTUS) organizovane po potrebi i direktno upotrebljive za izradu predmera i predračuna

PROSTORNI PRIKAZI

Vrlimo se problematici prikaza objekta Po mišljenju autora kao i mnogih arhitekata koji koriste CAD programe jedna od osnovnih prednosti korišćenja računara je to što je već u ranoj lazi projekta moguće sagledati prostorni izgled objekta kao i njegovog enterijera U tu svrhu AEC ima dva moćna oruzija komandu PERCEIVE i WALKTHROUGH

Komanda PERCEIVE je lak način da dobijete perspektivni prikaz objekta ili enterijera Njena velika prednosi u odnosu na klasičnu komandu DVIEW je u lakšci odredivanja očne tačke posmatraca Jednostavnim umetanjem simbola čoveka i biranjem tog simbole na upit *Select viewer* AEC generiše perspektivu sa kom posmatranja na mestu gde je simbol čoveka umetnul U sadejstvu sa proširenim mogućnostima *AutoCAD V11* ovo je mocno oruđje u razgovorima sa investitorima!

WALKTHROUGH konsli dobro poznati *Autoflix* za generisanje animiranog prikaza šetnje kroz objekal Grafički nivo prikaza je ne nivou mogucnosti EGA kartice ali je ono što se dobija za deselak minuta stvarni utisak kretanja kroz objekat koji može biti još na nivou idejnog projekta Sve što konsnik treba da odredi je putanja kretanja da li želi prikaz žičanog modela ili sa skrivenim linijema

Ovo što ste do sada pročitali mozda previše lepo zvuči i zato ćemo navesti i neke nedostatke ovog zaista odlicnog modula Sve navedeno važi dok je geometrija vašeg objekta „u okviru normale" Kompilkovanij prostorni oblici se ipak moraju modelovati iz *AutoCAD-a* Problemi se javljaju i prilkom nekontrolisane upotrebe pomašanih *AutoCAD* i AEC komandi Na osnovu iskustava u radu sa verzijama uz *AutoCAD V10* i *V11* i sagledavanjem napretka u razvoju programa podseticemo da je ovo lek verzija 2 11 Ovaj modul u svakom slučaju zasluzuje da mu se „proglada kroz prste" jer su mu prednosti brojnije od mana Jedna od najvećih mana u naŠim uslovima je to što za ozbiljan i profesionalan rad zahteva vrlo jak (čitati skup) računar kao vodilom I *AutoCAD* Ali to vec nije knvrka Autodaska Uz dalji razvoj AEC ce sigurno postati standard za CAD u oblasti arhitekture na PC računarima

Snimci uz ovu rubriku

Enterijer kafe poslastičarnice objekta BAS
Autor **arh M Đordevic**
Projektant **arh V Čitlučanin**
Prostorni model u AECu **erh Đ Grujić**

Kontakt adresa:

OSA računarski inženjering
Šumadijski trg 6a Beograd
Tel/Fah 559-845

LOTUS ZA SIROMAŠNE

U prvoj polovini ove godine svetu su predstavljena najnovije verzije najpopularnijeg tabelarnog programa svih vremena, koji može da se pohvali sa preko 14 miliona registrovanih korisnika!!!. Nakon verzije 3.1+, koja radi samo na AT mašinama, Lotus je izbacio i verziju 2.3 koju možete pokrenuti na svemu što se zova PC i koja ima neke jedinstvene mogućnosti među tabelarnim programima.

Vladimir Stamenović

Program 1-2-3 verzije 2.3 stiže u kutiji sa izmenjenim dizajnom, ali sa standardnim sadržajem. Moj primerak je „evaluation copy" sa disketama od 5.25" 360 KB (9 kom.) i 3.5", 720 KB (5 kom.). Odmah pada u oči promenjen dizajn uputstava za rad. I sadržaj je poboljšan. Od odličnih uputstava, na koja sam već navikao, napravljena su još bolja. Odmah sam primetio da su stranice u knjigama obeležene redom „kako je Bog rekao" a ne po poglavljima. Time je bar meni olakšano pretraživanje i sadržaja i knjiga.

PRE NEGO ŠTO POČNETE

Prvo uputstvo koje treba pogledati je „Before You Begin – Start Here", gde je ukratko dat sadržaj paketa i neophodna objašnjenja o potrebnom hardveru i OS-u da bi 1-2-3 mogao da radi. Među knjigama je i jedan dodatni papirić, kasnije se ispostavilo da ga treba obavezno pročitati, naslovljen kao „1-2-3 Release 2.3 and Expanded Memory Software". U njemu su nabrojani svi *expanded memory* programi koje ova verzija 1-2-3 podržava. Ispostavilo se da neće da radi sa mojim OEMM 386 verzija 4.0 već je zahtevao verziju 5.0. U svakom slučaju, provereno je da li je vaš *expanded memory* softver na listi podržanih. Ako nije, moraćete da potražite adekvatnu zamenu. U vezi sim, moram da napomenem da mi uopšte nije jasno zašto verzije 1-2-3 2.x rade sa *expanded* memorijom, a verzije 3.x sa *extended*. Ako ništa drugo, sa konsničke strane gledano, ako upotrebljavate obe verzije, malo je nepraktično.

Dalje, od uputstava za rad, tu su i neizbežne informacije za one koji unapređuju staru verziju novom. „Quick Start for 1-2-3 Upgraders" (60 strana). Za nova konsnike uputstvo za instalaciju programa dato je u knjizici „Getting Started" (50 strana). Kompletne informacija sa svim neophodnim detaljima o programu 1-2-3 i svim dodatnim (Add-In) programima date su u korisničkom vodiču „User's Guide" (400 strana). Detaljno uputstvo o raspolozivim funkcijama i makro komandama naći ćete u „Functions and Macros Guide" (200 strana). Naravno, tu je i kratki pregled svih komandi i njihov raspored po menijima „Quick Reference" (45 strana). U paketu su priložene i nalepnice sa nazivima funkcijskih tastera i to za obe vrste tastatura.

Što se hardvara tiče, 1-2-3 verzija 2.3 radi na IBM PC/XT i AT računarima sa najmanje 384 KB RAM-a. Ako ćete raditi i sa WYSIWYG-om onda je minimum 512 KB. Obavezan je čvrsti disk na kome morate imati slobodna bar 2 MB za 1-2-3. Za sve dodatne programe obezbedite još 5 MB. Nakon učenja možete obrisati programe za obuku i tako osloboditi 2 MB. Od softvera je potrebna DOS verzija 2.1 ili viša ili *Windows* verzija 3.0.

Moj primerak nije zaštićen, i ako to nisu i normalni prodajni primerci, onda je to lepa novost od Lotusa. Instalacija je uobičajena, sa „Diska 1" se startuje program Install koji Vas vodi do kraja. Traje toliko koliko je potrebno da se otpakuje i prebaci na disk oko 6.5 MB datoteka. Najviše vremena treba za generisanje fontova, mada to dosta zavisi i od brzine razunara. Preporučujem da za početak generišete samo bazične fontove, a kasnije i ostale.



Nemed Marjanović

SNAGA U DODACIMA

Nakon pregleda sadržaja (instalacijom) stvoreni direktorijuma uočljivo je bilo da uz 1-2-3 ide i nekoliko dodatnih programa (datoteke .ADN). Čini se da Lotusova strategija razvoja programa 1-2-3, bar kako je ja vidim, podrazumeva maksimalnu kompatibilnost sa prethodnim verzijama, povećanje brzine rada, unapređenje načina končenja memorije, kao i uvođenje neophodnih izmena i poboljšanja u programu koje zahtevaju korisnici, a da sve dodatne mogućnosti i druga znatna poboljšanja idu preko takozvanih Add-In (dodatnih) programa koji se pozivaju i koriste po potrebi. Takvom strategijom obezbeđeno je da 1-2-3 i dalje ostane najjednostavniji i najbrži tabelarni program.

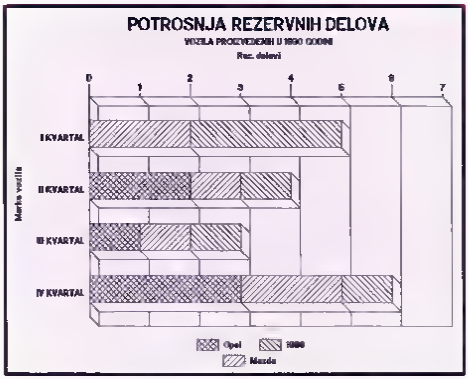
Od dodatnih programa koji, inače dobar program, čine znatno boljim, lakšim i konsnijim, tu je *WYSIWYG* koji omogucuje izuzetno kvalitetnu prezentaciju podataka iz tabele, *Auditor* kao dodatak programu pomoću koga se vrlo tako prate međusobne zavisnosti formula i otkrivaju greške, *Viewer* omogucuje pregled tabela pre učitavanja, izuzetno jednostavno linkovanje celije iz tekuće i neke druge tabele na disku, kao i još neke pogodnosti. Tu su i *Macromanager* i *D360* koje smo već videli u prednodnoj verziji. Treba još napomenuti da se uz ovaj paket isporučuje vrlo kvalitetan program za učenje 1-2-3, smešten u podirektorijum TUTOR, a nazvan *1-2-3-Go!* Ako vam nije neophodan, ne morate ga instalirati na disk, jer zauzima oko 1.2 MB. Tu je i program za obuku za rad sa WYSIWYG-om koji se zove *Wysiwyg-Go!* u podirektorijumu WYSYGO (303 KB na disku).

Prema nekim mojim informacijama, bilo je planova da se ovi programi za učenje, kao i Help, prevedu na naše jezike, ali kako sam kasnije saznao, od toga, za sada, na žalost, nema ništa. Prava šteta, jer bi to bilo od koristi velikom broju potencijalnih konsnika koji ne barataju dobro engleskim jezikom. Po odustajanju Lotusa od takvih poslova može se zaključiti da naše maleno i razbijeno, takoreći nepostojeće tržište tek treba da doživi informatički bum. Primera radi, evo jednog podatka kod nas je do sada prodato par hiljada paketa 1-2-3, više nego i jednog drugog, dok je, recimo, samo jedna engleska kompanija (British Gas) nedavno kupila 5 000 kopija!

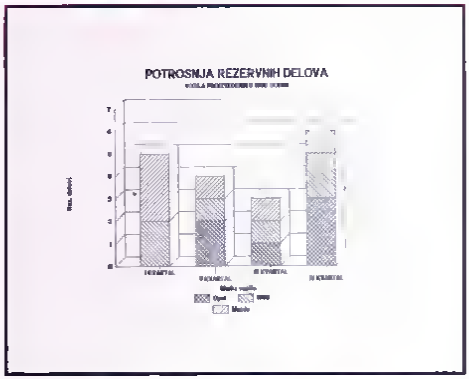
OSLIKANE TABELE

Nakon prebacivanja programa sa diskete na disk i njegovog startovanja, odmah pada u oči novi dodatak programu *WYSIWYG spreadsheet – publishing* – dodatni program koji predstavlje značajno unapređenje u odnosu na prethodni *Always*. Kao što verovatno znate, WYSIWYG (izgovara se kao vizvig) je skraćenica od „What You See Is What You Get", odnosno, „ono što vidiš na ekranu to dobiješ na štampaču". Prvi put je u *Lotusu* viđen u 1-2-3 ver. 3.1 i evaluirao je iz dodatnog programa koji se zvao *Impress*. Evo ga sada i u sastavu mlađeg brata, čime su se smanjile razlike među verzijama 1-2-3 za „sve" i 1-2-3 za IBM PC/AT računare. *WYSIWYG* omogucuje kreiranje fantastičnih tabela, što se izgleda bče, i to potpuno jednakog ne ekranu i štampaču. U startu 1-2-3 je podešen da automatski učita *WYSIWYG*. Naravno ako vam to ne odgovara možete ga isključiti, odnosno pozivati po potrebi. Po isključenju *WYSIWYG*-a 1-2-3 na ekranu izgleda kao stan dobro poznati Lotus s tim da se oslobađa oko 100 KB RAM-a. Izveštaji pravljeni sa 1-2-3 i *WYSIWYG*-om po kvalitetu se mogu meriti se onima rađenim u specijalizovanim grafičkim i izdavačkim programima.

Među pogodnostima koje možete konstiti spadaju i crtanje okvira ili senčenje ćelija ili oblasti ćelija, promena tipa, stila ili veličine slova, ubacivanje grafikona u tabelu ili maksimalno dotenvanje izgleda grafikona. *WYSIWYG* se startuje učitava, preko /Add-in menija. Kada je u memoriji pokreće se preko sopstvenog menija odnosno pomocu ( ) ili ako imate miša pomeranjem kursora u prostor iznad tabele. Prebacivanje između *WYSIWYG* i 1-2-3 menija vrši se pomocu desnog tastera na mišu odnosno Esc sa tastature.



Primer jednostavnog 3-D grafikona sa horizontalnim .



..i vertikalnim šipkama



Slika 1

Primer izveštaja sa mešanom grafikom i tekstom

Konsnici prethodne verzije 1-2-3 2.2 koji su navikli na *Allways* ne treba da brinu. *WYSIWYG*-ov meni je vrlo sličan prethodniku i on je u potpunosti kompatibilan sa *Allways* (ALL) fajlovima makroima i printi bibliotekama. *WYSIWYG* sadrži 4 Bristream tipa fontova Courier Swiss Dutch i Xsymbol a korisnik može birati veličinu od 3 do 72 tačke. Moguca je upotreba osam različitih fontova istovremeno, s tim da se mogu napraviti mnoge različite varijante koje ćete čuvati u posebnoj biblioteci fontova i pozivati po potrebi. Veoma je interesantna i upotrebljiva mogucnost da se sačuva neki format pod svojim imenom. Na primer, ako često koristite Swiss tip fontove u veličini od 14 tačaka, na lamino plavoj pozadini sa žutim tekstom (to mi se dopada), onda ćete, koristeći komandu Named Style (toj kombinaciji dati ima i lako ćete je pridružiti bilo kojoj ćeliji ili oblasti kada zatreba.

Brojne su druge mogucnosti koje možete koristiti promena boje pozadine pointera negativnih brojeva, celija sa skinutom zaštitom okvira itd. Izbor broja redova koji se vide na ekranu, pa čak i jednostavni tekst procesor. Osim toga, može se ubaciti i grafikon (PIC fajl) ili neka slika (CGM metafajl) u tabelu. I ne samo to. Vrlo su velike mogucnosti dotjerivanja i ispravki slika i grafikona kroz meni koji je dosta sličan onome u *Lotus Freelance* grafičkom paketu. Grafik se može poboljšati na primer dodavanjem teksta, linija, strelica, pravougaonih ili kružnih-elipsastih elemenata ili simbola. Tu su i mogucnosti pomeranja zakretanja ili rotacije kao i promene veličine grafika-slike. Što se liče boja na raspolaganju su 224 nijanse koje možete izabrati kroz meni.

je neke stvari lakše uraditi mišem. Na primer, jednostavniji je poziv menija kao i prebacivanje iz Lotusovog i *WYSIWYG* menija izdavanje komandi označavanje oblasti ćelija kao i kretanje po tabeli. Sa leve strane tabele posloji 5 simbola – ikona pomocu kojih je jednostavno kretanje po tabeli u svim pravcima kao i pozivanje Help-a.

Kad sam već kod Help-a, moram reći da je znatno unapređen u odnosu na sve prethodne verzije računajući i 3.1. Izuzetno je „konteksi senzitivan". U svakom momentu rada pritiskom na taster F1 dobićete pravu informaciju i pomoc. Help prozor ne pokriva ceo ekran, a na kontrolnom panelu može se videti sadržaj aktivne ćelije. Kretanje po Halpu je jednostavno i jasno posebno ako radite sa mišem. U njega su uključeni i svi dodatni programi. Pri radu sa mišem jako mi se dopada način izdavanja komandi. Naime, za razliku od drugih poznatih programa ovog tipa, pri radu sa 1-2-3, levi taster na misu je Enter a desni Esc. Takva kombinacija omogućuje komforan rad i brzo prilagodavanje radu sa mišem.

Inače u ovoj verziji 1-2-3 sva podešavanja (Settings) idu preko dijalog boksova koji sadrže veliki broj mogućnosti koje se mogu upotrebiti a smanšlene su na jednom ekranu. Promenu u podešavanjima možete vršiti i preko tastature i mišem. S tim da je rad sa mišem dosta zgodniji i brži. Dovoljno je da kursor nameste na odgovarajucu opciju i kliknete levim tasterom (Enter). To najčešće zamenjuje izdavanje nekoliko komandi kroz meni. Upotreba misa je naročito zgodna kada radite sa *WYSIWYG*-om gde vrlo jednostavno možete na primer menjati tip slova čak lakše nego u 1-2-3 verzija 3.1 ili 3.1+.

NONTEKSNI HELP

Lotus 1-2-3 verzija 2.3 podržava rad sa mišem, bilo da radita sa *WYSIWYG*-om ili bez njega. Stari korisnici koji su navikli da rade preko tastatura neće imati nikakvih problema jer i dalje egzistira čuveni Lotusov meni sa poznatim rasporedom komandi. Medutim, verovatno ćete se, vremenom, uveriti da

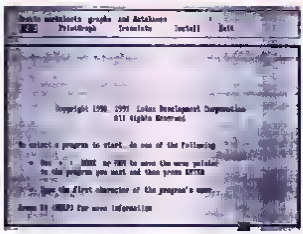
VIEWER I AUDITOR

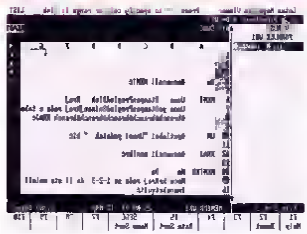
Sledeći dodatni program koji mozemo koristiti u 1-2-3 je *Auditor*.

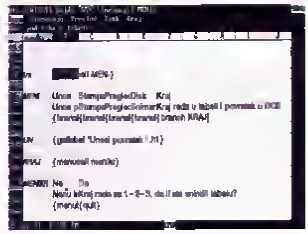
Isti je i u verziji 2.3 i u verziji 3.1+. Ocenjujem ga kao izuzetno korisnu novinu u programu. Mnogi korsnici 1-2-3 koji su pravili velike i kompleksne tabele znaju za probleme koji se ne retko ieda javljaju. Naime, mnoge formule u tabeli su u medusobnoj vezi, tako da im se često „gubi" trag a nisu retka ni situacije kada sa javi CIRC (circular references) indikator. Naći početak i praiti razvoj zavisnosti formula i funkcija kroz tabelu a, pogotovu naći uzrok pojave CIRC indikatora je prava robija. Sada sa *Auditorom*, to je daleko lakše. On ima svoj sopstveni meni u Lotusovom stilu kroz koji je vrlo jednostavno naci poziciju svih formuka u tabeli sve formule u tabeli koje zavise jedna od druge pa i redosled kojim preuzimaju podatke jedna od druge! Osim toga lako ćete naći onu ili one celije koje su svojim sadržajem uspostavile „kružni lok" formula. Još je jednostavnije reditl sa mišem kroz dijalog boks. Dakle, radi se o izuzetno korisnom dodatnom programu koji znatno olakšava kreiranje i prepravku radnih tabela posebno onih komplikovanih i velikih.

Još jedna velika novost i veoma konstan dodatni program je *Viewer*. Prvi put se pojavio u vec poznatom Lotusovom *utility* programu koji se zove *Magelan*. Ako ste zaboravili naziv tabela koju treda da učitate, ne morate koristiti neki fajl editor kao sto je onaj Nortonov ili slican, da biste videli sadržaj, pa po njemu ime tabele. Dakle, tak nakon loga mogli ste da startujete 1-2-3 i vašu tabelu. Slična situacija je i kada ste hiali da kombinujete više tabela u jednu. *Viewer* je rešio te probleme na veoma efikasan i racionalan način. Znaci bez napuštanja 1-2-3 startovanjem *Viewer*-a mozete pregledati sadržaj svih WK1 WKS i WRK datoteka i zatim onu koju želite učitati u 1-2-3.

Da spomenam još jednu od *Viewer*-ovih mogucnosti koja olakšava povezivanje aktivne i tabela na disku. Naime u 1-2-3 je i u prethodnoj i u ovoj verziji predvidena mogucnost da u aktivnoj tabeli koristite podatke iz tabela sa diska. Dakle, da povezujete više tabela u jednoj. To je jako zgodna i korsna stvar, ali muka je u tome što kad vam je zatrebao podatak iz tabele na disku, morali ste snimiti tekucu – radnu tabelu, pa učitati tu drugu sa diska da biste videli adresu ćelije sa potrebnim podatkom za







pamtiti adresu pa ponovo učitati tabelu na kojoj ste radili da bi ubacili podatak. Razumljivo je da je i pored koristi koje donosi linkovanje tabela bilo dosta posla da se to uradi kako treba. No u ovoj verziji, *Viewer* čini taj posao takvim pa i zabavnim. Naime, kad ge aktivirate sa leve strane ekrana vidite spisak raspoloživih tabela na disku, a sa desne sadržaj. Kada dodate do odgovarajuće tabele, strelicom u desno uđete u tabelu. Kretanje po tabeli je isto kao i u 1-2-3. Dođete do odgovarajuće ćelije i pritiskom na Enter završavate posao. *Viewer* automatski ubacuje vezu i adresu odabrane ćelije sa diska u aktivnu tabelu. Sjajna stvar koja će zasla pomoći pri pravljenju radnih tabela. Jedina moja zamerka odnosi se na brzinu rada. *Viewera*. Naime, svoja probe sam radio na XT laptopu sa taktom od 10 MHz i za svako linkovanje trebalo mi je nekoliko sekundi. Sa manjim tabelama to ide nešto brže. Na slarom 386/16 računaru se linkovanje obavlja dosta brže, ali ipak nedovoljno brzo, pogotovu kada se navikna na izuzetnu brzinu samog 1-2-3.

ZGODNA POBOLJŠANJA

Ovi dodatni programi su izuzetno korisni i jednostavni za upotrebu, ali nisu i jedine novosti odnosno poboljšanja u ovoj verziji 1-2-3. Recimo, zgodna stvar je uređena sa tastarom Delete (razlikujte ga od Del). Kada pritisnete taj tastar, brišete sadržaj aktivne ćelije! Dakle, nije više potrebno pritisnuti 4 tastera (/**R**ange, **E**N-**TER**) da biste obrisali sadržaj ćelije. Preporučujem vam da, dok se na naviknete, držite uključenu UNDO opciju za svaki slučaj. Una predena je /**Range Search** komanda jer sada u string koji tražite možete uključiti i džokere (? *). koji funkcionišu na isti način kao i u DOS-u. Što dopunosi većam komloru u radu.

Ima poboljšanja i u slilnicama koje život znače. Kada hoćete da prekinete sa radom u tabeli (**/Worksheet Erase** ili **/Quit**) 1-2-3 će vas upozoriti da nista snimili tabelu na disk. Isti slučaj je i sa komandom **/File Retrieve**. Što ocenjujem kao izuzetno korisno jer znam za mnogobrojne situacije kada se radi jedna tabele pa učita druga a da prelhodna nije snimlja na. U tim slučajevima nema pomoći podaci su izgubljeni.

Još jedne slitnica mi se čini korisnom u pravljenju aplikacija. Naime, dosta je poboljšan način rada sa oblastima ćelija. Može se na primer **Range** označiti pre nego izdamo komanda koja nesto reda s njim. Oblasi se može označiti sa lastalure sa F4 ili mišem (pritisnete levi taster, držite ga i pomerate miša).

Već smo rekli da kroz *WYSIWYG* ima dosta poboljšanja u radu sa grafikonima i grafičkom prezentacijom podataka. Osim toga tu su i dva nova tipa grafikona: HLCO (high – low – close – open) i *Mixed* (mešani). Oba ova tipa smo videli u 1-2-3 verzije 3.1 i 3.1+, kao i u *Symphony*. HLCO tip grafikona ima neke veze sa prikazivanjem podataka sa Iržišta, ali sa nikad nisam poludio da stvarno shvatim njegovu namenu. *Mixed* grafikoni kombinuju *Bar* (šipkasti) i *Line* (linijski) prikaz podataka. Osim toga tu je i popunjavenje prostora ispod linija u grafikonu prikazivanje Bar grafika horizontalno ili vertikalno, kao i kvazi 3D grafik. Ni ova varzija populamog programa name pravu 3D grafičku prezentaciju podataka. Jednom sam već negde napisao zašto je to tako. Iako u programima ovog tipa drugih proizvođača redovno možeta naci ovu mogucnost. Lotus je i dalje nema. Kako sam saznao osnovni razlog je jedno istraživanja koje tvrdi da 3D grafika nije najbolji način prezentacije podataka (mada meni deluje najelraktivnije) a i korisnici na insistiraju na toj mogućnosti. Ako je to tako, onda je besmisleno ugrađivati u program nešto što nije neophodno, a poskupljuje proizvod i sigurno ga dodatno komplikuje. Da li je to tako prosudićete sami.

ŠTAMPANJE TABELA

Ima novina i kod štampanja tabela. Po ugledu na varziju 3.x dodale su mogućnosti štampanje *Encoded* datoteka, i štampanja „u pozadini". Ova poslednja je izuzetno korisna jer omogućuje nastavak rada u tabeli dok printer štampa. Ono što mi tu smeta je da se to mora predvideti pri startovanja 1-2-3. Naime ako hocete tu mogucnost morete prvo pokrenuti program BPRINT pa tek onda 1-2-3 u okviru koga treba odabrali **Bacground** opciju iz **Print** menija. Povoljno je to što BPRINT zauzima samo 6 KB RAM a ako vam i to zalrca ba moraćete da reaalujete računar. Dakle reći je o veoma korisnoj mogućnosti koja je mogle biti malo spretnije uradena. Ovda treba napomenuti i da je dodato dosta novih drajvara ze različite tipove printera. Za štampanje grafikona zadržana je slara koncepcija koja predvida posaban program *PrintGraph* koji služi isključivo za štampanje grafikona. Dakle kada napravite neki grafikon, morate ga snimili na disk, izaći iz tabela, aktivirati *PrintGraph* i onda štampati.

Čak je i dizajn nepromanjen u odnosu na slarure verzije. Igrom slučaje moj kolega sa posla je tu naišao na problem. Naime, napravio je grafikon sa jednom promenljivom ali sa 4500 ličaka. Kad je hteo da ga odštampa, 1-2-3 je prijavio gresku – poruku „ oul of memory". Eksperimentisanjem smo utvrdili da je sve u redu do 2500 ličaka. Isti kompleian grafikon uspešno smo štampali iz *Symphony* i 1-2-3 verzija 3.0. Zbog nedostatka vremena nisam stigao da konsultujem Lotus Support. Je nisam u stanju da dam objašnjenja za ovako nešto. No, nija to, verovatno neki veći problem. Ono što mi smeta je to da Lotus nija našao način da po ugladu na 1-2-3 ver. 3.x omogući štampanje grafikona iz jednog menija, dakle bez *PrintGraph* programa.

Za kratko vreme radio sam sa ovim paketom na četiri različita računera. Prvo o IBM PC/XT – ima 1-2-3 stvarno lapo radi na XT u sa 512 KB RAM-a. Brzina je bila u skladu sa hardverom, ali je bitno da nije bilo problema sa skrolovanjem ekrana koje me je lako bacalo u nesvesi kod *Quattro Pro*-a 1-2-3 podjednako dobro radi na desktopu sa Hercules karticom kao i na laptopu sa CGA monohrom displejem. Naravno nedostak memorija se brzo javlja kao ograničavajući faktor. Sa 512 KB ako koristite WYSIWYG, možeta pravit relativno male tabele, a nečele moći da koristite *Viewer* i *Auditor* jer nema dovoljno RAM-a. Zato ako već radite sa IBM PC/XT računerom sa 512 KB RAM-a preporucujem vam da dopunite RAM do 640 KB pa ćete tako moći da koristila sve pogodnosti koja nudi 1-2-3 verzija 2.3.

NA VIŠE RAČUNARA

Instalaciju sam izvršio i na AT 286/12 računaru sa VGA grafičkom karticom. Sve je dobro funkcionisalo. 1-2-3 je radio veome brzo. *WYSIWYG* takođe. *Viewer* sporo mada brže nego na XT-u. Na slarom 386/16/VGA sam bio najzadovoljniji. Sve je bilo kako treba. I brzina i količina memorija i boja. Međutim najbrlje od svega je da su se komande izvršavale na svim tipovima racunara, vrlo brzo. Sporosi koja mi je veoma smalala kod QP-a, u ovom programu na posloji te nija imalo smisla meriti vremana njihovog izvršavenja.

GUI koji je napravio Lotus je odličan. U retkim prilikama dolazi do skrolovanja ekrana koja se odvija podnošljivom brzinom. Iako da ne smela. Učitavanje i snimanje datoteka kao i samog programa sa/na disk je dosta brzo, znatno brže nego kod verzija 3.x. Unapređen je i makro jezik dodavenjem nekoliko novih makro komandi koje smo upoznali u verzijama 3.x a zadržana je i slopostolna kompatibilnost sa prelhodnim verzijama programa. Ukratko, glavne prednosti ove verzije Lotusa 1-2-3 nad drugim programima ove vrsle su brzina i jednostavnost u radu, kao i ogroman broj dodatnih programa ne Iržišlu koji znalno uvecavaju njegovu moć. O lome ćemo nešto vise već u sledecem broju.

U nedostatke bih ubrojao. In stvari nedostatak prave 3D grafike (kod nas je vole) štampanje grafikona iz posebnog programa i canu. Qna je za naš standard ipak visoka bez obzira što je reč o ovako kvalitetnom proizvodu. Po mom mišljenju definitivno najboljem na Iržištu.

Korisna adresa
M.P. YUSACQ
21000 Novi Sad
Bulevar radničke samouprave 1
Tel. 021/395-414, Fax 021/398-119

OBJEKTI NA KLIPERU

O objektnom programiranju u „Računarima" je pisano više puta. Na žalost, objavljeni tekstovi su bili previše formalni da bi „zagrejali" programere. Ipak, pažljiviji posmatrač može uočiti da se u objektnom programiranju iza glomaznih i komplikovanih definicija ipak krije nešto zaista praktično.

Nenad Bartolanin

Clipper 5.0 je već nekoliko meseci među nama. Sa soodm je doneo nekoliko veliklh novina, a svakako jedna od najinteresantnijih je novi tip podataka **objekti**. Na žalost objektno programiranje je prilično nepopularno kod nas. Zašto je tako, nije lako odgovoriti: možda se radi o skepsi ljudi koji na veruju previše u novotarije, ili je u pitanju loša obaveštenost. Iek pobornici objektnog programiranja se kod nas mogu prebrojati na prste. Ovaj tekst ima za cilj da donekle objasni šta je to objektno programiranje i mesto objekata u novom *Clipper*-u.

ZAŠTO OBJEKTI?

Objektno orijentisano programiranje je nov način za pisanje programa. Pojednostavljeno rečeno zasniva se na ideji da se **vernija** slika ono što program preuosi iz realnog (?) sveta. U suštini se pri pisanju programa vrši preslikavanje nekih manje ili više realnih odjakata u neku programsku konstrukciju - obično skup promenljivih i procedura. Program tako **opisuje** realan objekat i ima za cilj da **modelira** naka njegova svojstva. Na primer, ako pišemo program za evidenciju radnika, mi na neki način modeliramo sva radnika predstavljajući neke njihove važne osobina nizovima promenljivih. Pri tom modeliranju skoro da nemamo ograničenja jedini zahtev je da taj model funkcionše. Otud i tolika raznolikost rešenja svakog, pa i najjednostavnijeg zadatka: svaki realni objekat predstavlja na sebi najbliži način.

Objektno programiranje pokušava da u programe uvede mnogo vernije slike realnih objekata u vidu novog tipa podataka koji se takođe naziva **objekt**. Promenljiva tipa objekt može „sadržati" u sebi promenljive svih ostalih tipova uključujući tu i druge objekte - slično klasičnim slogovima. Kao dodatak može sadržati i **funkcije** koja se odnose na taj objekt. Takav pristup omogućava u startu mnogo jednostavniju i čitljiviju strukturu programa.

Još jedan važan činilac iz života je pranel u programe **razmena informacija**. Svaka aktivnost se u suštini može svesti na neku razmenu informacija - objektno programiranje omogućava da i vaši odjekti u programu tako funkcionišu da primaju i šalju informacije i da izvršavaju određene akcije na osnovu toga. Vaš program ima koordinatorsku ulogu i on kreira i briše objekte, šalje im i prima od njih razne poruke.

Šta se u stvari time dobija? Pretpostavimo da pišemo program za kontrolisanje figura na šahovskoj tabli. Da bismo imali potpuno određenu figuru, moramo uvesti neka promenljive u kojima ćemo pamtiti tekući status figure. Uzećemo za primer piona. Uvešćemo promenljive **X** i **Y** u kojima ćemo pamtiti tekuće koordinate piona, logiciu promenljivu **Ulgri** koja pokazuje da li je figura u igri, promenljivu **Boja** koja određuje odju figura. Napisaćemo procedure za kretanje piona napred za uzimanje protivničke figure levo ili desno. Na primer, početak takvog programa moza sa napisali kao na slici 1.

Primetite da su ovde promenljive **X**, **Y**, **Ulgri** i **Boja** razdvojene od procedure **Napred** bez stvarnog razloga, u stvari, samo zato što paskal zahteva da se odvojeno deklarišu promenljiva i procedura. Ova promenljive i procedure zajedno čine programsku „sliku" realnog **objekta** piona na šahovskoj tabli. Dalje navedana promenljive su globalna i može ih nesmetano menjati ma koja procedura, možda i greškom – ničim nije označeno da samo procedura koja sa odnose na piona mogu menjati ova promenljive. Ako nam zatreba da napravimo kontrolu na samo jednog piona, već svih figura na šahovskoj tabli, naš polazni program će morati da trpi velike izmena: sva promenljiva će morati da postanu nizovi, a procedura će dobiti još po jedan argument, kako bi se znalo na kog sa piona/figuru odnosi pomeranje.

Dalje uopštavanja ide još teže. Ako nam zatreba da program kontroliše više tabli, verovatno će nam biti jednostavnija da ponovo napišemo ceo program nego da prepravljamo postojeći. Pregledniost takvog programa će se takođe veoma smanjiti. Iskusniji programeri će već u početku „objedlniti" srodna promenljive u jedan slog i tako povećali preglednost programa i olakšati modifikaciju. Postavlja se pitanja zašto na bismo i proceduru **Napred** na neki način dodelili tom slogu, kad ona po prirodi stvari tamo i pripada? Jedini odgovor je – zato što nam to jezik na omogućava! A objektno programiranja nam upravo to omogućava: da grupišamo promenljive i procedura opisujući tako mnogo prirodnije realne odjekte. Kao što lepo piše u uputstvu za *Turbo Pascal*, objektno programiranje nam omogućava da razmišljamo na prirodniji način – kao slikar koji slika objekat!

OBJEKTIVIZIRANI CLIPPER

Na žalost, *Clipper* nije postao pravi objektno orijentisan jezik (zlobitici će reći: hvala bogu!). Međutim, u verziji 5.0 uveden je tip podataka **object** i definisani su odjekti u nekoliko krtičnih podsistema u **Browse**, **Get** i **Error** podsistemu. U okviru ovih sistema postoji definisano nekoliko klasa objekata i moguce je samo kreiranje i „upravljanja" objektima iz tih klasa – ne postoji nikakva mogućnosi kreiranja novih klasa. Ipak i postojanje samo ovih „ugrađenih" klasa i operacija sa njima omogućilo je veliki korak napred ranji glomazni i komplikovani sistemi zamenjani su mnogo fleksibilnijim i moćnijim mehanizmom. Jedini sistem koji je iz nepoznatih razloga „zaobiđen" prilikom uvođenja objekata je **MemoEdil** funkcija za manjanje mamo polja i stringova izgleda da su objekti u novom *Cloper*-u samo proba za nešto mnogo krupnija – ako konsnici prihvate ove izmene, vrlo verovatno će *Clipper* uskoro dobiti ponu objektnu implementaciju!

Novi sistemi su uradani vrlo oprezno – skoro sva alara funkcija su zadržane zbog kompatibilnosti i mogu se nesmetano koristili. Među tim, sve su označena kao zastarela i ne preporučuju se za upotrebu.

BROWSE SISTEM

Svi korisnici *Clipper*-a sigurno dobro poznaju funkciju **DBEdit** – to je veoma korisna funkcija koja omogućava tabelarni pregled podataka iz jedna ili više datoteka. Na zadatom delu ekrana ispisuju sa podaci u obliku tabele: korisnik može da pomera kursor i tako bira slog i polje na koja će primeniti narednu operaciju. U ulazne agrumante funkcije spadaju osim koordinala prozora i simodli u zaglavlju podnožju, izmedu kolona itd. Reklo bi se da ništa na nedostaje. Ali stvarnost je vrlo surova: takva **DBEdit** funkcija ne bi mogla da zadovolji ni osnovna potrebe. Njena prava moć proizilazi iz fleksibilnosti: ako promenom argumenata funkcije pri pozivu ne možete da postignete željeni efekat, postoji mogućnost definisanja korisničke funkcije koja se izvršava uvak kada se pritisne neki tastar. U toj funkciji, na primer, možala definisati akciju kad korisnik pritisna naki funkcijski tastar. Posle izvršenja akcije nastavlja se normalno izvršavanje funkcija **DBEdit**. Ovakav sistem je uglavnom dobro funkcionisao, ali bio je daleko od savršenog. Setila se samo egzibicija kad je potrebno osvetliti ceo tekući slog a ne samo jedno polje (kad treba „zamrznuti" neku kolonu i slično. Uočila da ovde svi problemi nastaju uglavnom zato što **DBEdit** nezavisno obavlja funkcije za koje bi više odgovaralo da ih sam korisnik kontroliše, na primer ispisivanja ekrana i slično. Ali, kad bi se korisniku dozvolila kontrola ekrana, sa prednostima bi došla i povecana obaveza oko tehničkih detalja. Tako je **DBEdit** funkcija uvek bila rasligrnuta između dve suprotna koncepcije: crna kutija na koju korisnik ne može da utiče – skup funkcija niskog nivoa dostupnih korisniku, ali korisnik vodi računa o tehničkim detalijima. Nijedan problem ovog tipa nije jednostavno rešiti. Dosta dobri kompromisi su mogući i postojeća implementacija **DBEdit** funkcija to i potvrđuje.

U novom *Clipper*-u problemu se pristupa sa potpuno druga strane. Kada vam zatreba da vršita tabelarni pregled kreiraćeta specijalni **TBrowse** objekat, koji predstavlja zamenu za **DBEdit** funkciju. **TBrowse** objekat je u stvari skup promenljivih i funkcija koja opisuju trenutno stanje prikaza na ekranu. Neka od tih internih promenljivih možemo samo čitati neke možemo i menjati, podešavajući tako rad **TBrowse** objekta. Funkcijama koje sadrži **TBrowse** objekat narediujemo razne akcija: regenerisanje ekrana, promenu boja i slično. Ovakve funkcije se u objektnom programiranju zovu **metodama**. U stvari kompletna komunikacija između našeg programa i **TBrowse** objekta svodi se na razmenu raznih informacija (**slanje i primanja poruka**). Komunikacija sa objektom se izvodi operatorom „." (dvotačka). Na primer, ako želimo objektu po imanu **TB** da „naredimo" da regeneriše ekran, napisaćemo

```
TB:refreshAll ()
```

Ako želimo da saznamo tekuću poziciju **TBrowse** kursora, napisaćemo

```
Pozicija := TB:colPos
```

Ovakva koncepcija ima daleko više prednosti nego što bi se moglo pomisliti. Počev od mnogo

Listing 1

```
VAR N, T : INTEGER;                    (* Statusne promenljive figure *)
    Uigri : BOOLEAN;
    Boja  : (crna, bela)

    I, J, K : INTEGER;                 (* Pomocne promenljive programa *)

PROCEDURE Napred (K : 1..8);           (* Procedura koja se odnosi na pione *)
BEGIN
  IF BoleNapred (N) THEN
    T := T + N
  END
END;
```

U klasičnim programskim jezicima procedure su potpuno razdvojene od promenljivih koje im pripadaju

Listing 3

```
#include "Inkey.ch"

PROC Main
   LOCAL TB1, TB2

   // Otvaranje datoteke
   USE Psimenl INDEX primer NEW
   USE Psimenl INDEX primer NEW

   @ 2,2 TO 24,79 DOUBLE | @ 1,30 TO 11,30 DOUBLE

   // Kreiranje prvog objekta
   TB1 := TBrowseDB (1,1,23,38)
   TB1:addColumn (TBColumnNew ("",{|| o_p1}))

   // Kreiranje drugog objekta
   TB2 := TBrowseDB (1,40,13,78)
   TB2:addColumn (TBColumnNew ("",{|| o_p1}))

   WHILE .T.

      // Ispisivanje ekrana se odvija paralelno
      WHILE !TB1:stable .OR. !TB2:stable
         IF !TB1:stable
            Select (1)
            TB1:stabilize()          // Prvi objekat
         ELSE
            Select (2)
            TB2:stabilize()          // Drugi objekat
         END IF
      END DO

      // Ceka na akciju
      Ch := Inkey(0)

      DO CASE
      CASE Ch == K_ESC .OR. Ch == K_ENTER          // Kraj rada

         // Pomeranje dole prvog objekta
         CASE Ch == K_F1 ; Select (1) ; TB1:down()

         // Pomeranje dole drugog objekta
         CASE Ch == K_DOWN ; Select (1) ; TB2:down()

         // Pomeranje gore prvog objekta
         CASE Ch == K_F2 ; Select (1) ; TB1:up()

         // Pomeranje gore drugog objekta
         CASE Ch == K_UP ; Select (2) ; TB2:up()

         // Sinhronizovano pomeranje dole
         CASE Ch == K_F3 ; Select (1) ; TB1:down() ; Select (2) ; TB2:down()

         // Sinhronizovano pomeranje gore
         CASE Ch == K_F4 ; Select (1) ; TB1:up() ; Select (2) ; TB1:up()

         // Sinhronizovano pomeranje u suprotnim smerovima
         CASE Ch == K_F5 ; Select (1) ; TB1:Up() ; Select (2) ; TB2:Down()

      END CASE
   END DO

   CLOSE ALL

RETURN
```

Primer rada sa više *TBrowse* objekata

Listing 5

```
PROC Main
   LOCAL OldH, x

   // Postavljanje nove funkcije za obradu greske
   OldH := ErrorBlock ({ |Err| ErrFunc (Err)})

   // Greska!
   x := 8 / 0

   ? "Kraj programa"

   // Vracanje normalnog handlera
   ErrorBlock (OldH)

RETURN

// Funkcija za obradu greske
PROC ErrFunc (NO)
   IF Alert ("GREŠKA : " + NO:description, {"Kraj", "Greška"}) == 2
      QUIT
   END IF
RETURN .T.
```

Obrada grešaka korišćenjem *Error* objekata

Listing 2

```
#include "Inkey.ch"

PROC Main
   LOCAL TB, Ch

   CLS
   // Otvaranje datoteke
   USE Strmenl NEW

   // Kreiranje novog, praznog TBrowse objekta
   TB := TBrowseDB (0,0 24,78)

   // Dodavanje kolona TBrowse objektu
   TB:addColumn (TBColumnNew ("IME I PREZIME", {|| Prezime}))
   TB:addColumn (TBColumnNew ("ADRESA", {|| Adresa}))
   TB:addColumn (TBColumnNew ("TELEFON", {|| Telefon}))

   // Postavljanje razmik parametara
   // Separatan izmedju naslovlja i tela
   TB:headSep := "-"
   // Broj zamrznutih kolona
   TB:freeze := 1

   // U ovoj petlji se vrsi ispisivanje ekrana
   WHILE .T.

      // Vrsi se ispisivanje ekrana , ove dok se ne ispise ekran !!!
      // dok korisnik ne pritisne neki taster
      Ch := 0
      WHILE !TB:stable .AND. Ch == 0
         Ch := Inkey ()
         TB:stabilize ()       // Ispisi jedan red ekrana
      END DO

      // Ako je ekran ispisan, a korisnik nije nista pritisnuo,
      // ceka se komanda!
      IF TB:stable .AND. Ch == 0
         Ch := Inkey(0)
      END IF

      // Analiza pritisnutog tastera
      DO CASE
         // Kraj
         CASE Ch == K_ESC ; EXIT

         // Gore
         CASE Ch == K_UP ; TB:up()

         // Dole
         CASE Ch == K_DOWN ; TB:down()

         // Levo
         CASE Ch == K_LEFT ; TB:left()

         // Desno
         CASE Ch == K_RIGHT ; TB:right()

      END CASE
   END DO

   CLOSE Primer1

RETURN
```

Jednostavan program koji omogućava listanje datoteke na ekranu korišćenjem objekta *TBrowse*

Listing 4

```
#include "Inkey.ch"

PROC Main
   LOCAL GO, X := 114

   CLS

   // Kreiranje get objekta
   GO := GetNew ()

   // Koordinate get objekta
   GO:row := 2 ; GO:col := 10
   // CodeBlok za menjanje izmedju objekta i bafera
   GO:Block := {|v| IF (PCount() == 1, X := v, X)}

   // Aktiviraj objekat
   GO:setFocus()

   WHILE .T.
      // Ispisi trenutnu pozicilju kursora
      Xr := Row() ; Xc := Col()
      @ 24,0 SAY GO:pos
      KeyPos (Xr, Xc)

      // Ceka komandu
      Ch := Inkey(0)

      DO CASE
         // Quit
         CASE Ch == K_ESC ; EXIT
         // Exit
         CASE Ch == K_ENTER ; GO:assign () ; EXIT
         // Levo
         CASE Ch == K_LEFT ; GO:left ()
         // Desno
         CASE Ch == K_RIGHT ; GO:right ()
         //Upisivanje slobodn
         CASE IsAlpha (Chr(Ch)) ; GO:insert (Chr (Ch))
      END CASE
   END DO
   // "Ispusi" objekat
   GO:killFocus()

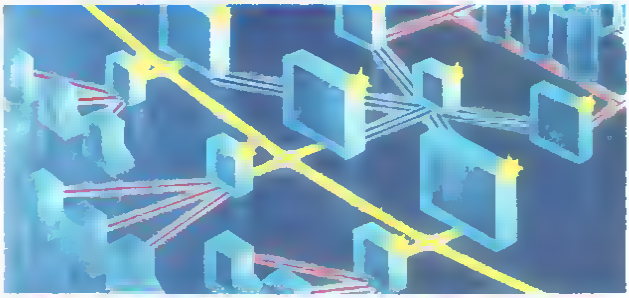
   // Stampaj promenljivu
   QOut (X)

RETURN
```

Program vrši unos vrednosti promenljive *X* bez poziva funkcije *ReadModal*

većih mogućnosti prikaze pa do jednostavnijeg pisanje programe. Ako vam se u početku učini da su ovakvi programi komplikovaniji gledajte. Uostalom, skoro sve je jednostavnije od funkcije sa 18 argumenata!

Na slici 2 je prikazan jednostavan program koji omogucava listanje datoteke na ekranu. Posle otvaranja datoteke kreira se novi objekat za pregled datoteke i dodeljuje se promenljivoj **TB**. To se radi funkcijom **TBrowseDB()** koja kao argumente ima koordinate prikaza na ekranu. Na ekranu se još uvek ne vidi nika-



kva izmena. Svaki **TBrowse** objekat se sastoji iz nekoliko **TBColumn** objekata (svaki ob njih opisuje jednu kolonu tabelarnog prikaza). Ovo vam se može učiniti kao nepotrebno kompliko vanje ali sačekajte sa tim je i stvarna situacija la kva. Tabelarni prikaz se sastoji ob prikaza kolo na **TBColumn** objekat se kreira pozivom tun kcije **TBColumnNew**. Na primer, sledeći po ziv

Column := TBColuranNew („ADRESA", {|| Adrasar->adresa})

kreira lakav objekat i dodeljuje ga promenljivoj **Column**. Pri kreiranju treba zadati laksi koji će se ispisati u zaglavlju kolone („ADRESA") i kodni blok (*code block*) koji služi za izračunavanja vrednosti. U našem primeru je to vred nost polja **adresa** u radnoj oblasti **adresar**. Kada se kreira **TBColuran** objekat treba ga do deli nekom postojecam **TBrowse** objektu. To se radi konsčenjem funkcije (meloba) **addCo luran**. Ne primai

ŤB addColumn (Column)

Time smo postojećem objektu **TB** dobali jednu kolonu **Column**. Potpuno analogno dobajemo i ostala kolone. Mnoga interna podešavanja se vrše izmenama internih promenljivih **TBrowse** objekta. Na primer, ako želimo da u prikazu prva kolona bude zamrznuta, napisaćemo

TB·freeze := 1

Ispisivanje ekrana se obavlja pod potpunom kontrolom korisnika što je značajna razlika u odnosu na **DBEdit**. Svaki poziv funkcije **stabilize** ispisuje po jedan red ekrana (prikaza). Ka da ceo ekran bude ispisan, interna promenljiva **stable** imaće vrednost **TRUE**. Znači, dovoljno je da pozivamo funkciju **stabilize** u petlji sve dok objakat **TB** ne postane „stabilan". A šta ako korsnik pritisne neki taster pre nego što se ekran potpuno ispiše? **DBEdit** funkcija je nepravedno postupala: ako se pritisne neka njena komande usred ispisivenja ekrana, ispi sivanje se prekida i obrađuje se zadata ko manda. Ovo je logično: zasto ispisivati do kraje tekuću stranu eko korsnik u međuvremenu za traži prelazak na novu stranu! Ali, zašto to ne bi radilo (a ne radi) i sa komandama koje uvodi korisnik? U **DBEdit**-u je to bilo nemoguce a sa **TBrowse** sistemom je trivijalno. Naime, po što se ispisivanje ekrana obavlja pod kontro lom našeg programa, dovoljno je prekinuti ispi sivanje kad se pritisne bilo koji taster i nastaviti ga tek posle obradana komande. Zatim se je dnostavno analizira pritisnuti taster i preduzime odgovarajuća akcije. Za jednostavne akcija poput pomeranja kursora polja postoje ugradane funkcije (melobe). Ne primer, ako želimo da pomerimo kursor u levo, napisaćemo

TB left ()

Za razne druge efekte imate na raspolaganju veliki spisak ugrađenih funkcija pomoću kojih možete potpuno da kontrolišeta ceo **TBrowse** objekat i njegova kolona. Ovde se mnogo lak ša kontrolišu pojedinačne kolone, jer su ona posebni objekti. Na primer, jednostavno je pro menniti boju svake kolona posebno i slično. Jedna od osnovnih karakteristika gde se mo žda najbolje vidi pradnost ovakvog pristupa je rad se više **TBrowse** objekala. Naime, kada se pozova funkcija **TBrowseDB** kreira se je dan poseban **TBrowse** objekat nezavisan ob već postojecih njega možale potpuno obvoje no kontrolisati. Potreba za ovim se javlja prilič no često, zato je i **DBEdit** funkcija bila ieentralna, ali to nikako nije omogućavalo paralel nu kontrolu dva prikaza. Sa objektima je to skoro trivijalno. Na slici 3 je prikazan lakav pri mer. Primetite da broj prikaza koji se istovia meno kontroliše ničim nije ograničen. Čak i pro gram koji ih kontroliše uopšte nije složen.

Sve u svemu, prilično je teško zamisliti neki problam tabelarnog prikaza koji sa pomoću **TBrowse** objekata ne može rešiti. Varovalno će naredne verzije *Clipper*-a imati proširen i poboljšan ovaj sistem, ali i za sada je izvanre dan. Jedina zamerka koje bi sa mogla uputiti je prilična tromost celog sistema.

GET SISTEM

Stari **GET..READ** sistem za učitavanje po dataka bio je u većini slučajeva sasvim dovo ljan za realizaciju svih maski za učitavanje po dataka. Nedostajalo je nekoliko mogucnosti za potpunu kontrolu, ali je u *Clipper*-u 5.0 situacija poboljšana opcijama **COLOR** i **WHEN**. Za jednički problam je što je **READ** naredba ogra ničavala kontrolu programa na funkcije nave dene u **VALID** i **WHEN** opcijame programa tu jednostavno nije imao nikakvog pristupa. Novi **GET** sistem ja otišao korak dalje.

Učitavanje jedne promenljive se opisuje po moću tzv. **GET** objekata. Svaki objakat iz te klase sadrzi informecije koje su neophodne da bi se odredilo učitavanja jedne promenljive koordinate na ekranu, tekucu vrednost **VALID** i **WHEN** funkcije itd. Kada se u programu na vede niz **SAY-GET** naredbi pretprocesor ih prevodi u poziva odgovarajućih funkcija koje kreiraju niz **GET** objekata i podesavaju im razne parametre. Naredba **READ** se prevodi u poziv funkcije **ReadModal** koja kao argument prihvata niz **GET** objekata i omogućava nam da kursorom biramo promenljivu i menjamo njenu vrednost. Funkcija **ReadModal** je real izovana kao *Clipper* program i proizvoljno se može menjati. Međutim, ne postoje nikakve prepreke da ne iskonstimo objekte i funkcije iz **GET** klase i tako sami realizujemo neki specifi čan unos podataka. Na slici 4 je prikazan pro gram koji vrši unos vrednosti promenljive **X** bez poziva funkcije **ReadModal**.

Prvo se kreira **GET** objekt **GO** pozivom fun kcija **GatNew**. Zatim se podešavaju razni para metri kreiranog objekta: pozicija na kojoj ce se izvršiti učitavanje i slično. Potrebno je i postavi ti internu promenljivu **block** koja sadrži kodni blok koji vezuje promenljivu čija se vrednost učitava i bafer u kome se vrednosi unosi. Taj blok treba, kada se izvisi bez argumenata da vrati tekuću vrednost promenljive koja se učita va, a kad se pozove sa argumentom da dodali zadatu vrednost promenljivoj koja sa učitave. Aktiviranje unosa se počinje funkcijom **setFo cus** koja računa **Eval (blok)** i dodjenu vred nost upisuje u bafer za unos podatka. Zatim se viši komunikacija sa objektom **GO** veoma slično kao se **TBrowse** objektom čeka se da korisnik pritisne taster i na osnovu njega pre duzima odgovarajuća akcija. Za standardne akcija posloje gotova funkcije tipa **left**, **right** i sl. Primetite da su lo potpuno razlicite funkcije ob onih kob **TBrowse** objekala, iako imaju isla imena – ovde su deo **GET** objekala. Tekuci status učitavanja nam je potpuno dostupan za hvaljujuci internim promenljivim unutar objekta **GO**. Tako se u detom primeru uvek ispisuja tekuća pozicija kursora u zadnjoj liniji ekrana. Naravno ova akcija može biti mnogo složenija i svrsishodnija – važno je da ovakav sistem omogućave potpunu kontrolu. Ako radite na ovom nivou možete veoma jednostavno realizovell svoj **GET** sistem sa sasvim različitim svojstvima ob standaidnog sa drugom kontro lom boja i kursora itd.

ERROR SISTEM

I novi **Error** sistem je uvrstio objekte u svoj standaidni repertoar. Raniji postupak u slučaju greške u programu bio je sledeci: u zavisnosti ob tipa greške pozivala se obgovarajuća proce dure koja bi na osnovu broje linije koda gre ška i sličnih podalaka preduzimala odgovaraju cu akciju. Kontrola grešaka za programera se svodila na izmenu funkcije za obgovarajuću obradu grišake. Ako bi vam zatrebala različita kontrola u raznim delovima programa, postu pak se svobio na zamorno manipulisenje glo balnim indikalonima što nikako nije zabavan posao.

Sada je ceo postupak mnogo prirodniji. Ka da se u sistemu dogodi greška, kreira se **Error** objekat. On sadrži sve informacije o tome gde je i kako i koja griska nastala. Ovaj objekat se zatim prosleđuje kao argument kodnom bloku za obradu greška koji standaidno ispisuje po ruku i prekida program. U proizvoljnom delu programa mi možemo obradu greške usmeriti na naš kodni blok pozivanjem funkcije **Error-Block** koja postavlja novi kobni blok za obradu greške. Na slici 5 je prikazan primer u kome se obrada greške usmerava na funkciju **Er rFunc** koja dalje obrađuje grešku. Ukoliko želi te samo privremeno da zamenite standardnu obradu grasake obavezno sačuvajte stari kod ni blok za obradu greške i kasnija ga vraiite pozivom funkcije **ErrorBlock**.

Objektno programiranje je veoma popular no u inoslranstvu i skoro svi noviji programski jezici se ne neki način proširuju objektima. Na ravno i najbolje stvari imaju smisla tek ako počnu de žive u nečijim programima. Mora se pomenuti da naši programeri nisu uvek usvajali najbolje – njihovo ponašanje je vrlo često bilo neočekivano. Ali ne može se poreci da su uvek imali izoštren sluh za dobre stvari i zato ne treba previše sumnjati da će objekti vrlo br zo postati sastavni deo njihovih programa. Uo stalom ne postoji ni jedan razlog da to ne bu de tako!

BAZE NA C-U

Osnovnu snagu *C*-a, osim veoma elegantnih jezičkih konatrukcija, čini proizvod *Code Base 4* firme *Sequiter Software* namanjan radu sa bazama podataka.

Nenad Batoćanin

Code Base 4 je biblioteka koje sadrži sve funkcije neophodne za rad se bazama podataka i to sa datim izvornim kodom. Značajan podatak je da su funkcije kreirane uglavnom tako da se oponašaju naredbe *dBASE*-a, tako da oni koji znaju *C* i *dBASE* ili *Clipper* programski jezik neće imati većih problema oko upolrebe funkcije *Code Base*-a. Naravno zbog specifičnosti *C*-a nije bilo moguće jednostavno „prepisati" sve naredbe u *C* funkcije tako da će vam izvesten period prilagođavanje biti potreban. Primenjeni koncepti su uglavnom jednostavni, tako da se možete vrlo brzo navići na njih. Paket sadrži i nekoliko lepo urađenih demo programa, iz kojih se može puno toga naučiti.

RAD SA BAZOM PODATAKA

Sve funkcije su podeljene u grupe po operacijama koje obavljaju, a svake grupe ima karakterističan prefiks. Na primer, sve funkcije koje se odnose na rad sa bazom podataka počinju sa **d4** (**d4use**, **d4close**, ...). Kao što je već rečeno, većina funkcija je urađena tako da podseća po nazivu i sintaksi na odgovarajuće naredbe *dBASE*-a. Na primer ako želite da otvorite DBF datoteku, napisaćete **d4use** („**PROBA**"). Ne žalost, poljima ne možete tako jednostavno da pristupite kao u *dBASE*-u. Nelme, prvo se mora izvršiti učitavanje tekućeg sloge u memoriju, pa se tek onda pristupa vrednostima. Slično je postupak kad treba izmeniti slog. *dBASE* stvara iluziju kod programera da su mu polja tekućeg sloga direktno dostupna, baš kao i normalne promenljive, dok operacije se pokazivačima obavlje internio. Slična je situacija i sa pristupanjem radnim zonama datoteke indeksirana rid.

Pristupanje u *Code Base*-u se izvodi pomoću identifikacionih brojeve (*reference number*). Na primar. pomoću funkcije **f4ref** dobijamo ne osnovu imena polje broj koji mu je dodeljen e zatim konščenjem tog broja obavljamo operacije sa tim poljem. Na primer ako želimo da dobijemo tekuću vrednost polja **NAZIV**, napisaćemo

```
f4str(f4ref("NAZIV"))
```

U suštini rad sa ne razlikuje previše – razlika je oglavnom na nivou teksta programa. jer *dBASE* „krije" suvišne poziva za postupanje raznim objektima. Ne slici 1 je prikazan primer u kome se na ekranu ispisuju sve vrednosti polje **A_NAZIV** iz detoteke **ARTIKLI**. indeksirane po Iom polju. Da maksimalno ubrzavanje programa poziv **f4ref** se može izbaciti izvan petlje, jer uvek vraća istu vrednost. Za upisivanje vrednosti u daloteku (ekvivelent naredbe **REPLACE**) koriste se funkcije sa prefiksom **f4r**. Na primer, ako želimo da promenimo vrednost polja **NAZIV** tekućeg sloga napisećemo

```
f4r_str (f4ref ("NAZIV"), „Nova vrednost");
```

Ako je polje drugog tipa koriste se funkcije **f4r_int**, **f4r_double** itd. Ako pokušate da upišete neku vrednost u polje neodgovarajućeg tipa, funkcija će automatski izvršiti konverziju. Teko će se, na primer, ako upišete celi broj **123** u polje **CHARACTER** tipa, izvršiti automatske konverzija broja **123** u string „123"

Svaku otvorenoj datoteci se dodeljuje bafer za tekući slog u kome se čuvaju vrednosti polja tekućeg sloga. Čitanje vrednosti iz datoteke i upisivanje u ovaj bafer obavlja se uvek kada se pozove neka funkcija za pomeranje pokazivača sloga. Na primer **d4skip(2L)** pomara pokazivač za dve sloga napred i automatski učitava vrednosti polja tekućeg sloga u bafer, odakle dalje možete čitati vrednosti. Samo otvarenje datoteke funkcijom **d4use** ne **učitave** vrednost prvog sloga – to se mora uraditi eksplicitnim pozivom funkcije **d4top**. Omogućen je i alternativni (na nižem nivou) prastup bazi podataka, eksplicitnim navođenjem naredbi za učitavanje i upisivanje sloga.

Značajna je i mogućnost veome finog podešavanje komunikacije programa sa DBF datotekama podešavanje veličine komunikacionog bafera i slično. Ukolilko ne koristite ove funkcije *Code Base* će konstiti podrazumevene vrednosti koje su uglavnom dobro izabrens za standardne primene. Podešavanje daje rezultate samo u slučaju veoma malih ili veoma velikih datoteka i kada se datolekama prislupa na neuobičajen način (strogo sekvencijalno ili sa učitava veliki broj malih rasutih slogova).

Postoje sve neophodne funkcije za kreiranje kopiranje indeksiranje i listanje datoleka za postavljanje filtera i sl. Posebno je elegantan rad sa filterima. Pozivom funkcije **x4filter(usr_func)** postavlje se filter koji računa korisnikove funkcije **usr_func**. Ako ta funkcija vrati 0, slog je vidljiv, a u suprotnom (1) slog je nevidljiv. Zatim se funkcijama **x4top**, **x4bottom**, **x4skip** i **x4seek** obavlja sasvim standardan rad sa datolekom. Za svaku datoteku se vodi liteta funkcija koje vrše filtriranje. Tako se dodavenje dodatne funkcije za filtriranje po nekom uslovu obavlja još jednim pozivom funkcije **x4filter**. Ako je potrebno neutralisati filter, to se radi ili pozivom **x4filter_reset** (poništavaju se svi filteri) ili **x4filter_pop** kojim se neutrališe poslednje funkcija specificirane pomoću **x4filter** poziva. Slično je rad i se višestrukim relacijama između raznih datoteka/tabela.

Memo polje su podržana uobičajenim funkcijama se jednim izuzetkom. Ne postoji funkcija tipa **MemoEdit** za menjanje memo polja već se za to koristi neki eksterni program (editor), što je veliki nedostatak.

FUNKCIJE ZA KAČUNANJE dBASE IZRAZA

Run-time izračunavenje izraza nepoznatih u vreme prevođenja apsolutno je neophodno za svaki sistem koji redi sa bazom podataka. Takva potreba jevlja se kod raznih zahteve, počev od interpretiranja korisničkih upita, pa do izračunavanje vrednosti indeksa. *Code Base* ima modul koji je zadužen ze računanje *dBASE* izraza. Računanje se izvodi u dve faze. Prvo se zadati izraz interno prevodi u pseudo-kod pozivom funkcije **e4parse**. Zatim se funkcijom **e4exec** „izvršave" prevedeni izraz. Ove dva funkcije su objedinjene u funkciji **e4eval** koja odmah i prevodi i računa izraz. Ako se računanje izvodi u petlji, bolje je prvo izvršiti prevođenje, a zatim u petlji pozivati funkciju **e4exec**. Podržane su sve osnovne operacije i relacije (+, -, *, /, **, konkatenacija stringova =, <=, .OR. .AND. i sl.) i standardne *dBASE* funkcije (**CtoD**, **IIF**, **RecNo**, **Str**, **Upper**, ...). Moguće je i jednostavno dodevanje funkcija koje korisnik sam definiše. Kao elemente izraza možete koristiti i naziva polje tekućeg sloga, s tim što onda takav izraz važi do zatvaranja datoteke – pri ponovnom otvaranju datoteke polja dobijaju nove ID brojeve na osnovu kojih se vrši pristup vrednostima. Na primeru sa slike 2 računa se maksimalna vrednost izraza za celu datoteku. Izraz se prvo „prevodi", a zatim se posebno računa za svaki slog datoteke.

KORISNIČKI INTERFEJS

Funkcije korisničkog interfejsa u *Code Base*-u urađene su sa velikom pažnjom. Kao i funkcije za rad se bazom podataka, urađene su po enelogiji sa naredbama/funkcijama *dBASE*-a i *Clipper*-a, all postoji veoma močan skup funkcija koji ne postoji u *dBASE*-u i *Clipper*-u. Sve te funkcije su podeljene u nekoliko podsistema: sistem za tabelarni pregled (*Browse*) rad sa prozorima, menijima i unos podataka.

Za tabelarni pregled i izmenu podataka postoji posebna **b4browse** funkcija. Ova funkcija (osim što omogućuje standardne operacije tabelamog pregleda) dozvoljava i definisanje korisničkih funkcija koje zamenjuju fiksne delove **b4browse** funkcije. Na taj nacin se omogućava prilagođavanje pregleda specifičnim potrebama slično kao kod *Clipper* funkcije **DBEdit**. Sistem daluje dosle kompletno i pouzdano, mada se ponekad javljaju izvesne greške u radu.

Kompletan ulaz i izlaz može se obavljati preko sistema prozora koji omogućava kreranje zaleta efektnih korisničkih interfejsa. Na ekrenu se mogu otvoriti više prozora a za svaki pojedinačno se podešavaju razni parametri boje, okvir, natpis, ... Koordinate koje se upotrebljavaju za pozicioniranje kursora su relativna u odnosu na gornji levi ugao tekućeg prozora. Kada se aktivira naredni prozor, sistem sam vodi računa o restauraciji ekrane i ispisivanju aktiviranog prozora. Na raspolaganju je veliki broj funkcije počev od kreiranja i aktiviranja prozore pa do raznih funkcija za ispisivanje podatake i direktog upisa podataka u ekransku memoriju. Korišćenje svih funkcije je vrlo jednostavno.

Za unos podateke na raspolaganju su sistemi za definisanje menija i maski za unos podataka. Radi se slično kao u *Clipper*-u prvo se reznim pozivima definisu atavke menija i razni opcioni parametri, a zatim se poziva funkcija koje omogućava korisniku da izabere neku vrednost. Kod unosa podataka slično se definišu promenljive koje se unose, zatim **PICTU-**



Brzineki testovi *Code Base* va *Clipper*

Listing 1

```
#include <stdio.h>
#include <d4base.h>

main ()
{
   int i;
   int ind_ref;

   // USE Artikli INDEX Artikli
   d4use ("ARTIKLI"); i4open ("ARTIKLI");

   // LIST Artikli->a_naziv ALL
   for (i = 1, d4top (); ! d4eof(); d4skip (1L))
      printf ("\n %4d. %s", i++, f4str (f4ref ("A_NAZIV")));

   d4close_all ();
   return (0);
}
```

Program ispisuje na ekranu sve vrednosti odabranog polja iz datoteke indeksirane po tom polju

Listing 2

```
#include <stdio.h>
#include <d4base.h>

int main ()

{
   char *exp_ptr = "A_KOL * (A_PRODC - A_NABC)";
   double *val_ptr;
   char *exe_ptr;
   double max = 0;

   d4use ("ARTIKLI");
   e4parse (exp_ptr, &exe_ptr);

   d4top ();
   while (! d4eof ()){
      val_ptr = (double*) e4exec (exe_ptr);
      if (*val_ptr > max) max = *val_ptr;
      d4skip (1L);
   }
   printf ("\nMAKSIMALNA ZARADA = %f", *val_ptr);
   d4close ();
   return (0);
}
```

Program računa maksimalnu vrednost izraza za celu datoteku, iz-raz se prvo „prevodi", a zatim se posebno račune za sveki slog datoteke

Listing 3

```
#include <stdio.h>
#include <d4base.h>
#include <w4.h>

main ()
{
   int w1_ref, w2_ref, choice;
   int x = 1;
   double y = 2;
   char *ch = "ABC/DEF";


   // Definisanje dva prozora
   w1_ref = w4define (10, 10, 15, 50);
   w4border (DOUBLE, F_WHITE);
   w4popup ();

   w2_ref = w4define (12, 12, 23, 58);
   w4border (SINGLE, F_WHITE);
   w4popup ();
   // Meni u prvom prozoru
   w4activate (w1_ref);

   m4 ("ŠTAMPANJE");
   m4 ("ARHIVIRANJE");
   m4 ("IZLAS");
   choice = e4nctlvate (-1);

   // Zatvaranje prvog prozora
   w4deactivate (w1_ref);
   w4close (w1_ref);

   // Ucitavanje podataka u drugom prozoru
   w4activate (w2_ref);

   w4centre (1, "Ovo je centriran tekst");
   g4int (2, 0, &x);
   g4double (3, 5, &y); g4picture ("$$$$$$.##");
   g4 (4, 0, ch); g4picture ("AAA/AAA");
   g4read ();

   // Zatvaranje drugog prozora
   w4deactivate (w2_ref);
   w4close (w2_ref);

   return (0);
}
```

U programu se koriste neke jednostavne mogućnosti sistema za definisanje menija i mskai za unos podataka

RE stringovi boje za unos i slično... e zatim se poziva funkcija **g4read** koje učitave podatke Svi ispisi su u odnosu na aktiviran prozor a koordinate se mogu i automatski računati što omogućave veoma kvalitetan rad Ne slici 3 je primer u kome se koriste neke jednostavne mogućnosti ovog sistema

PRILAGODAVANJE

Veoma je važno da su sve funkcije iz biblioteke date sa izvornim tekstom programa na *C*-u ako vam u bilo kom trenutku zatreba da promenite neku od funkcija koju koristite, to možete učiniti ne sevršeno legalan i jednostavan način Ipak, menjanje funkcija nema neke svrhe ako ste početnik priložene funkcije realizuju veoma složen sistem, tako de je izmena bez detaljnog poznavanja tog sistema potencijalno veoma opasna Međutim, ono što je neophodno, prilagođavanje za korišćenje na našem jeziku izvodi se dosta jednostavno Ako se pri kreiranju biblioteke navede opcija **/LANGUAGE** dobijene biblioteke biće prilagođene nemačkom jeziku To se svodi na promene raznih parametara e za nas su najinteresantnije promena naziva meseca i relativan redosled ASCII tabele Izmene naziva meseca je trivijelne dok je prilagođavanje sortiranja podataka nešto komplikovanije Naime unutar ASCII tabele može se izvršiti proizvoljno preslikavanje, ali slove sastavljena od dva simbola (LJ, NJ, DŽ) neće tako biti na svojim uobičajenim mestima. Rešenje je ili u promeni procedure za sortiranje ili pisanju pomoćne filter-funkcije koje preslikava slove od dva simbola u neke ređe korišćene simbole, čime eliminišemo „dvosimbolna" slove Za detalje pogledajte datoteke **b4 c c4.c ; p4mlec.c**

Postojanje teksta funkcija daje dve sjajne mogućnosti ovom proizvodu podrška raznih prevodilaca i operativnih sistema *Code Base* je pisan ne standardnom *C*-u i možete ga koristi ze bilo koji prevodilac na *DOS*-u koji podržave standardni *C* U uputstvu piše da takuće verzije podrževe verzije *Turbo C Quick C, Microsoft C, Zortech C++, Welcom C 386* I prevodioce ne *SCO Unix/Xenix* operativnom sistemu Nažalost, kao što znate, čak i na nivou *DOS*-a postoje razlike u prevodiocima koje čine da tole složenij program ne možete identično napisati za dva različita prevodioca Takve mesta postoje i u *Code Base* biblioteci (na svu sreću, dosta su retka) i one se korišćenjem makroa i uslovnog prevođenje menjanju u zavisnosti od korišćenog prevodioca Ne primer ako konstite *Microsoft C* funkcije treba prevesti sa **-DMSC** Na taj način veši programi (pod uslovom de poštuju ista previle o prenosivosti) ostaju netaknuti pri prelasku na drugi prevodilac

I, što je najlepše, na sličan način kompletnu aplikaciju možete prebaciti i na drugi operativni sistem što je veoma dragocena osobina. Doduše, uputstvo nagoveštava izvesne probleme pri ovoj operaciji Preporučuje se posebna verzija biblioteke za *Unix*, a kod nekih funkcija je naglašeno da ne rade na *Unix*-u), tako da bi i pored direktiva tipa **#ifdef UNIX** u biblioteci to trebalo isprobati na konkretnoj aplikaciji

Postoji dosta funkcija raskog nivoa koje se najčešće ne koriste direktno i koje se koriste u funkcijama višeg nivoa. Njihovo izdvajanje i dokumentacija može vem koristi eko samostalno menjate delove biblioteke

dBASE, CLIPPER I CODE BASE

Code Base koristi DBF datoteke u standardnom *dBASE* formatu, a mogu se birati *Clipper* (NTX) ili *dBASE* (NDX) kompatibilni indeksi Kakvi će se indeksi koristiti mora se specificirati prilikom kreiranja biblioteke Korišćenje ovog standerdnog formata omogućave vem da koristite svoje stare podatke bez ikakve konverzije, a pruže vem i mogućnost razmene podataka izmedu *dBASE* kompatibilnih aplikacija i vašeg programe Takode ne treba zaboraviti i veliku broj pomoćnih programa uradenih za *dBASE* koje možete nesmetano koristi.

Što se tiče skupa naredbi funkcije *Code Base*-a pokrivaju sve osnovne funkcije za rad sa bazom podataka. Korišćenje je uglavnom slično *dBASE* konceptu, tako de ze one sa iskustvom na tom polju većih problema nema. Sistem za rad sa ekranom, učitavanje podataka i tabelarni pregled (*browse*) je višestruko nadgrađen i omogućave izradu veoma efektnih korisničkih interfejsa Letimičnim prelistavanjem uputstva možete videti dobro poznate funkcije **d4use d4close, d4go d4skip,**

Nastavlja se na strani 34

RIZIK NESPRETNE UPOTREBE

Programi za stolno izdavanje pojednostavljuju i olakšavaju grafički dizajn ali i, kao i sve moćne alatke, nose sa sobom rizik nespretne upotrebe. Čak i iskusni grafičari teško odolijevaju iskušenju da na jednu stranicu nabacaju sve dostupne fontove i geometrijske oblike koje im nude DTP sistemi. Na koji način optimalno iskoristiti oruđa koja nam nude grafičke aplikacije, te kako dizajnirati privlačan grafički materijal? Na ovo, ali i na niz drugih pitanja, odgovorit ćemo u u ovoj seriji tekstova.

Robert Slavetki

Ne postoji nikakva mistarija u kreiranju privlačnih grafičkih objekata posebice ne u okvirima stolnog izdavanja. Dizajn je nešto što se može analizirati i usavršiti konsteći neka osnovna ali prije svega esencijalna pravila jednostavnost, kontrasi, organizacija i image samo su neka od njih. Prisljavajući sebe da svoje projekta stolnog izdavanja razvijala u okviru ovih principa kvalitela i privlačnosti vašeg rede znalno ce poresti

Ova čelin osnovna pravila možete konsiti ze poboljšavanje svih vrsla dokumenata stolnog izdavanja počevši od zaštitnih znakova i logotipova preko raznih publikacija (knjige brošure prospekti) i objekata razmjene (poslanice memorandumi pisma) sve do nejrezličitijih poslovnih obrazaca (računi predračuni, narudžbenice) pa čak i elemenala poslovnih prezentacija (karte i grafikoni) koje su kod nas ionako još uvijek premalo u upotrebi te će tek dobiti na svojoj vežnosli

U većini slučajeva primjena ovih principa neće zahtijevati značajnije rekonslrukcije vaše rada. Bitno je shvatili da se navjerovatna poboljšanja mogu ostvarti senjom relativno malih izmjena koja zajedno, lansformiraju vaš dokumenal ostvaren u stilu „žurim na ručak" u nešto zaista lascinanlno

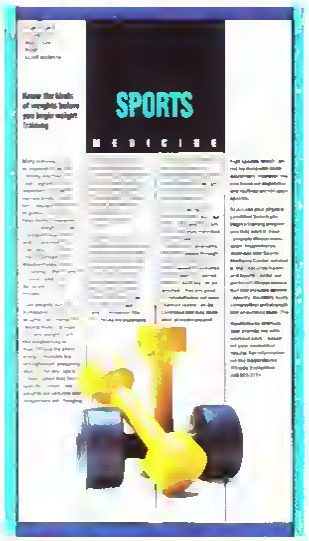
IDEJA

Iako ideja na spade u navedena pravila, ona je neizoslavno osnovna, i nelzbježne pretpostavka svakog rada. Prije nego što krenete u razradu ideje i „duboka razmišljanja" svakako se opskrbite neophodnim podacima kao što je količina teksta koja ulazi u dokument, broj slika vrsla papira na kojem će se izvršiti finalni otisak vrsta finalnog tiska (sve kopije bit će producirane na laserskom pisaču, olsel, silolisak) itd. Neizoslavno potražite i odgovore na pitanja popul U koju će svrhu služiti laj dokumenal, koja je primarna djelatnosi naručioca te što bi on naručenim dokumentom hbo naglasiti i drugo. U svakom slučaju potrudita se da imate što veću količinu ovih informacije

Razvoj ideje najbolje je započeti na papiru. Niti jedan računalni sistem još uvijek nije dostigao lu fleksibilnost koju posjeduje ruka u kombinaciji sa olovkom i papirom. Neizoslavno skiciraje osnovna oblika zamišljenog objekta ili stranice sve dok ne postanela zadovoljni ostvarenom razradom. Ukoliko posjedujete adekvatan skaner bit će vam od lzuzelne pomoći da skanirate ovu skicu i kasnije ju u nekoj DTP aplikaciji konstile kao podložak. Od log trenutka pa nadalje. Ia je skica vaša ideja. Nipošto ne odustajte od svoje prvolne zamisli već ju sprovedite do kraja. U lom će vam postupku svakako pomoći pravila koja slijeda.

PITANJE JEDNOSTAVNOSTI

Elektvan dizajn lemelji se na jednostavnosti. Vaši tiskani malerijali dobit će na svojoj kvaliteli toliko koliko čete bili sposobni odstra nili napotreban kič. Kič egzistira u raznim formama naglašenim stranicnim rubovima napotrebnim vertikalnim linijama izmedu susjednih kolumni, ili pretjeranom korišenju istih nječi ili fraze što je posebno uočljivo u nasiovima jednako kao i u naglašenom tekstu



Slika 1/a lako u pravilu treba izbjegavati upotrebu vertikalnih linija, ovaj primjer pokazuje da ne treba uvijek niti pravila uzimati kao isključivost. Vertikalne linije, koje se ne zaustavljaju samo na kolumnama već sa prolezu kroz cijelu stranicu, daju cjelom dokumentu određenu „vitkost" i dinamičnost.

Počnite sa pitanjem korisnosti okvira kao i vertikalnih la horizontalnih linija. Današnji DTP programi i aplikacija za obradu teksla omogućavaju da se na lagan i prelagan, način dodaju ovakvi geometrijski oblici. Kroničan slučaj okvirtisa najizraženiji je na novinskim stranicama. Ta su stranice prečasto oukvirene lanklm linijama (gotovo neuocljivim- ali one su tu lek toliko da se nađu)- zatim slijedi primarni naslov i pndruzeni mu leksi sa slikom sada već uokviren debalim naglašenim okvirom. Pecal sve mu lome daju popratni lekstovi često krači od 200 nječi naravno – uokvireni i smješteni u svoje „privatne kućice".

Linitis se najčešće pojavljuje kao popratna bolest zajedno sa okvirilisom. Najuočljivih simptom linitisa je preljerano dodavanje vertikalnih crta izmedu kolumni teksta. Ljudi najčešće dodaju ove linije jer nisu navikli na veci broj kolumni ili jednostavno zato što im alati u programima to omogućavaju. Rijedak je slučaj kada se ove linije dodaju uz stvarne potrebe (pogledajte sliku 1/a)

Prelјerano korištenje ovih alata rezultira rezlomljenom (fragmentiranom) stranicom. Umjesto da čitalac ima pred sobom jedinstvenu stranicu čiji bi sam oblik trebao asosirati na sadržej poruke on vidi serju pravokutnika linija i ostalih geometrijskih oblika. Ia tu i tamo koji takai

Jednostavnost također uključuje i optimalno korištenje ekrana (obojenih pozadina) jednako kao i obratnosti (npr bljeli leksi na crnoj pozadini). Ovi se grafički načini moraju koristiti izuzetno pažljivo. U prolivnom lekst na takvim podlogama može biti izuzelno necitljiv i odbojan

KONTRAST

Ovo može zvučati kontradiktorno ali jednostavnost može raditi zajedno sa kontrastom (pogledajte sliku 1/b) Eliminirajuci nepotrebne grafička elamente dobivale prostor neophoden za naglašavanje važnih objekata. Ovo će vam pomoći da izbjegnela problem tzv „sive" stranice. Siva stranica rezultat je nedovoljnog kontrasta. Sivi dukument prezentira čitaocu dosasnu jednoličnu masu teksta i udaljuje nas od početne točke da uhvelmo čitačevo oko i usmjerimo njegovu pažnju na sadržaj poruka

Jedan od najboljih načina da vizualno oživite i obogatite vašu publikaciju je da konstite kontrasine naslove. Naslovi su udica koja će zakačili čitaočevo oko i privući ga na vašu publikaciju. Što lakše čitalac locira uoči i pročita vaš naslov, lakše ce i s većom željom pročitati primarni teksl koji slijedi

Naslov možete učiniti kontrastnim tako de ga učinila znalno većim od pridruženog mu teksta. Dobra je također i ideja de se naslov zaokruži bijelim prostorom. Drugi način da istaknete naslov je taj da ga napišete u kontrasinom fontu. U jednoj od najčešće korištenih lehnika, serilni foni popul lajmsa konsti se za primarni teksl članka, dok se kontrasini bezserilni foni upotrebljava za naslove i podnaslove. Pogledajte sliku 1/c (Serifi su suptilni ornamentalni završeci na krajevima slova bez-serifni fontovi popul ovog u *Računarima*, nemaju ove završelke) Obzirom da serifi pomažu oku u prelazu s jednog slova na drugo i s jedne riječi na drugu serifna su pisma idealna za primarni taksi

Kontrasl se osim na leksl.u može primjenjivati i na ostalim grafičkim elementima. Zaokružujući grafikone, crteže i slika bijelim pro-

storom znatno ćete oplemeniti cjelokupan dokument. Slikama možete također dozvoliti da predu preko graničnih linija i okvira ili da, čak, prelaze do ruba stranice, i nikada na zaboravite bijelne nije nikada previše.

Jedan od najboljih načina da poboljšate izgled dvo-stranični dokumenata je taj da obje stranice sadrže isti jedinstven i dominantan vizualni element. To je najčešće grafički objekt, tako to može biti velik i izražajan naslov. Zapamtite da čitalac vidi lijevu i desnu stranicu kao jedinstvenu cjelinu (ili bi ih barem takve želio vidjeti), a ne kao razdvojene i nevezane dokumente. Susjedne stranice obavezno dizajnirajte kao cjelinu a nikada odvojeno a pri tome će vam znatno olakšati posao dvostrani 15" monitori popul NEC-ovog 5D, ili Apple ovog Two-Page monitora.

ORGANIZACIJA INFORMACIJA

Vašu ćete publikaciju znatno poboljšati ako informacije organizirate u manje grupe prikladne za manipulaciju i rukovanje. Izgled i čitljivost dokumenta znatno se povećavaju ako veliku količinu teksta prelomite u kraće segmente i pridružite im podnaslove. Takve podnaslove možete vizualno odvojiti od pripadnog im teksta okruzujući ih bijelim prostorom, koristeći kontrastne fontove, ili ih naglasiti sa jednostavnim grafičkim akcentom kao što je horizontalna linija. Veću količinu odnosnih informacija možete organizirati i tako da ulomcima pridružite strelice, zvjezdice (asterske), ili brojeve. Ovi će sitni objekti zasigurno privući pažnju čitaoca.

Ukoliko vaš dokument uključuje veći broj fotografija, nemojte ih nemilice razbacati po stranici već ih na neki način poravnajte i dovedite u sklad. U tu vam svrhu mogu pomoći alati koje sadržava gotovo svaka DTP aplikacija. ALLIGNMENT funkcije, ravnala (RULERS), te drjentacijske linije (GRID LINES).

SUBLIMINALAN IZGLED

Cjelokupna slika, odnosno image, kojim će se vaša stranica odlikovati ponajviše ovisi o fi-



Slika 1/b Iako može zvučati kontradiktorno, jednostavnost može ići zajedno s kontrastom, što najbolje prikazuje i ovaj plakat. Jarko žuta boja zvjezdica, te hladno plava boje pozadina totalno su kontrastne, stvarajući na taj način poseban kreativni efekt.

tografiji. Današnje Izište nudi nam enormnu količinu najrazličitijih pisama za stolno izdavaštvo, a svaka digitalna slovolivnica, eko imalo drži do sebe, ima u svojoj ponudi preko tisuću fontova. U takvom bogatstvu slovnih oblika zaista je teško odoljeti prekomjernoj upotrebi raznih pisama na jednoj stranici, tako da publikacije producirane na DTP sistemima prečesto sadrže prevelik broj fontova – stranica preraste u kič. Imajte uvijek na umu: nekad je kada potrebno upotrijebiti više od dvije familije fontova na jednoj stranici.


ACCENT

THE AFTERMATH OF DIVORCE

Tragic dog story provokes more snarls than sympathy

Slika 1/c Na ovoj je stranici naslov ispisan bez-serifnom Helveticom dok je tekući, primarni tekst, jednako kao i podnaslovi, ostvaren u kontrastnom, serifnom pismu. Ovom tehnikom, kao i činjenicom da je naslov „umočen" u bijeli prostor, dan je naglasak na nam naslov, a time i na temu poruke.

Neki fontovi nose u sebi modernu i prolaznu aromu dok drugi bude osjecaj starosti i sigurnosti. Ostale opcije uključuju formalni nasuprot neformalnom ili prijateljski nasuprot autoritativnom fontu. Ne možete, na primjer, koristiti isti font za izradu memoranduma poduzeću čija je djelatnost vezana uz pravo i financije te taj isti font iskoristiti za izradu plakata koji najavljuje gostovanje neke punk rock grupe. Zapamtite da svako pismo ima svoju osobnost svoj identitet kojime podražava i budi pojedine asocijacije kod samog čitaoca.

UPITAJTE SE

Pri analizi vašeg tiskanog dokumenta, u granicama gore navedenih pravila postavite sebi sljedeća pitanja.

JEDNOSTAVNOST

- Da li moje stranica posjeduje nepotreban kič?

Multi



Slika 1/d Kod dvostranih dokumenata obavezno smjestite jedan od dominantnih elemenata tako da zahvaća obje stranice. Na taj će način cjelokupni dokument dobiti na čvrstoći i kompaktnosti. U ovom na, konkretnom slučaju, glavni naslov „Multitalented" proteze cijelom horizontalom, a i slovke zabodene u metu naznetno prelaze na drugu stranicu.

- Jesam li prilikom dizajniranja prilagodio konstio akcente popul linija, okvira ili obojenih podloga?
- Koje nječi ili grafičke elemente mogu eliminirati kako bi osigurao dodatan prostor za naglašavanje bitnih objekata?

KONTRAST

- Da li moja stranica uključuje dovoljno kontrasta?
- Je li veličina naslova dostatna u odnosu na primarni tekst. I da li je – kada je to primjereno – naslov u kontrastnom fontu?
- Jesu li podnaslovi kontrastni sa pridruženim tekstom i fotografijama?
- Da li moja stranica sadrži dovoljno bijelog prostora?

ORGANIZACIJA

- Je li moja stranica organizirana tako da omogućava lako čitanje?
- Da li dvostrana ploha sadrži jedinstven, dominantan vizualni element?

IZGLED

- da li moj odabir fontova i stilova odgovara subliminalnoj impresiji te sadržaju, jednako kao i poruci koju bi dokumenat morao prenijeti čitaocu?

Prije nego što iskušate ova pravila na stvarnim dokumentima, primjenite ih na izmišljenim narudžbama prelamajući šuplji tekst. Na ovaj ćete način steći neophodno samopouzdanje i tehniku što će vam znatno pomoći u budućim ostvarenjima. Jednom rječju – eksperimentirajte.

U idućem broju posvetit ćemo se načinima korištenja jednako kao i adekvatnom odabiru programa za stolno izdavanje i grafički dizajn. Uz to ćemo prikazati ali i objasniti cjelokupan proces kreiranja zaštitnog znaka i logotipa od prvotne ideje pa do finalnog tiska.

Korisna adresa
DTP Studio NOVEL
41000 Zagreb, Milana Rešetara 44
Tel. 041/34 75 14 Fax 041/34 79 43

SKELETNI SISTEM

Skaletni aistam (aheil) je komandni intarfejs koji interpretira ulazne naredbe i aktivira odgovarajuće sistemaka programe koji čine jezgro operativnog aistama UNiX, a jezgro operativnog sistema (kernal) kontroliša pristup računaru, organizaciju mamorije, organizaciju datoteka, raspored rada procesa i raspodjelu sistemskih resursa.

Skeletni sistem je, u stvari, programski jezik sa varijablama, kontrolnim naredbama, potprogramima, prekidima i drugo. Organizovan je kao tumač (interpreter) naredbi, što znači da čita naredbu koju preko ekranskog terminala unosi korisnik. Interpretira je i preduzima sve potrebne akcije za njezino izvođenje. Kada je naredba izvedena, skeletni sistem daje informaciju korisniku (prompt) da je spreman prihvatiti sljedeću naredbu.

OSNOVNE OSOBINE

Skeletni sistem nije dio sistemskog jezgra, već korisnički program. Svaki korisnik može napisati svoj program koji će imati ulogu skeletnog sistema, međutim, poželjno je da to bude standardni program rasprostranjen na svim UNIX instalacijama, čime se postiže kompatibilnost rada na različitim računarima.

Za operativni sistem UNIX postoji više različitih skeletnih sistema, među kojima su dva najrasprostranjenija i najpoznatija. Jedan je razvijen u okviru firme AT&T (Bell Labs), zove se *Bourne shell* ili *sh* i uglavnom se koristi na AT&T verzijama UNIX-a. Poboljšana verzija tog skeletnog sistema se zove *Korn shell*.

Drugi skeletni sistem razvijen je na UCB (University of California at Barkeley) zove se *BSD shell*. *C shell* ili *csh* i rasprostranjen je na BSD verzijama UNIX-a. Na nekim UNIX instalacijama postoje oba skeletna sistema, pa svaki korisnik može izabrati se kojim želi raditi.

Oba skeletna sistema služe istoj namjeni, pa imaju i niz zajedničkih osobina, no *csh* ima neka dodatna, korisna svojstva koja *sh* nema. Najbitnija su: kontrola izvođenja procesa u pozadini i prednjem planu (background i foreground), povjest zadanih naredbi uz jednostavnu mogućnost njihovog ponovnog zadavanja, veći broj sistemskih poziva (za oko 60 više od *sh* skeletnog sistema) i drugo.

Skeletni sistem predstavlja pogodno sredstvo za razvoj ekspertnih sistema.

Osnovne osobine skeletnog sistema su:

— komandni jezik skeletnog sistema čita unesene naredbe na ekranskom terminalu računara i daje odgovor izvršavajući pozvana programe,

— skeletni sistem je programski jezik visokog nivoa. Isto kao što su FORTRAN, paskal, C ili bilo koji drugi. Napisani programi koji koriste ovaj programski jezik nazivaju se „shell scripts", procedure ili naredbe. Ovi programi mogu da se memorišu u datoteka i izvršavaju kao naredbe,

— skeletni sistemi obezbeđuju takvu okolinu, koja se može prilagoditi individualnim ili grupnim potrebama za rad sa spoljnim varijablama.

Skeletni sistem se aktivira automatski u trenutku uključivanja u računski sistem, a kasnije, po želji, naredbom *sh*. Uključivanjem korisnika u računarski sistem, započinje prvi proces skeletnog sistema aktiviranjem *login* programa i naredbi navedenih u datotekama *etc/profile* i *$HOME/profile*. Slijedeće naredbe skeletnog sistema kreiraju nove potprocese. Osim samog naziva programa ili naredbe, čiji je naziv naveden odmah poslije **sh**, nalaze se potrebni parametri ili varijable koje program zahtjeva.

Opšti oblik naredbe **sh** je:

sh (program) (parametri)

Pod programom u *sh* naredbi se podrazumjeva naziv naredbe ili programske datoteke koju treba aktivirati, dok se pod parametrima podrazumjevaju svi signali sata naredbi, kao i slijedeći

— **c** izraz — aktivira naredbe koje navodi izraz. Naredba **sh-c 'pwd I date'** aktivira naredbu **pwd**, a zatim **date**,

— **I** — skeletni sistem postaja interaktivni proces, što znači da će signal za prekid biti ignorisan,

— **r** — aktivira skraćenu verziju skeletnog sistema **(rah)**,

— **s** — ulazni podaci dolaze sa standardnog ulaznog uređaja, a izlazni (eko ne potiču od internih naredbi) odlaze na standardni izlazni uređaj za greške.

Naredba **sh** aktivira novi proces skalatnog sistema, čija će naredbe biti unijete sa tastature ekranskog terminala. Rad procesa sa prekida pritiskom na tastere <Ctrl—D> i kontrola se vraća procesu skelatnog sistema koji ga je započeo.

Naredbe koje skeletni sistem aktivira mogu biti zasebna i kombinovana, unešene direktno ili navedene u programskoj datoteci. Svaki red ulaznih podataka skeletni sistem analizira i raščlanjuje na naredbu (program) komandne parametre i nazive datoteka. Naziv programa se najprije poredi sa nazivima internih naredbi a zatim nazivima definisanih funkcija, jer ove dvije vrste ne zahtjevaju novi proces. U slučaju neuspjeha, skeletni sistem započinje novi proces i nastavlje traženja.

Naziv direktorija u kome se naredba nalazi može biti naveden kao dio njenog naziva ili izveden iz definicije PATH parametra. Prva nađena naredba se odgovarajućim nazivom se aktivira. Ako mod programske datoteka ne dozvoljava direktno aktiviranje naredbe, skeletni aistem kraira novi potproces. Naziv naredbe se unosi u posebnu kontrolnu tabelu zajedno se nazivom direktorija u kome se nalazi. Pri slijedećoj upotrebi iste nerebe, sistem najprije provjerava sadržaj tabela, pa tek ako naziv nije pnsutan provjerava PATH definicije. Nove naredbe mogu se dodati a stare ukloniti iz tabele internom naredbom **hash**, o čemu će kasnije biti govora.

PREUSMJERAVANJE I ULANČAVANJE

Procesima aktiviranim iz skeletnog sistema automatski se dodjaljuju tri otvorene datoteke:

— standardni ulaz,
— standardni izlaz,
— štandardni izlaz iz greške,

sa brojevima 0,1 i 2 koji adresiraju te datoteka. Pojam otvorena datoteka označava da određeni proces ima pravo vlasništve nad dotičnom datotekom. Datoteka 0 je otvorena za čitanje, koristi se kao standardni ulaz, i obično je to tastatura ekranskog terminala. Datoteka 1 je otvorena za pisanja i koristi se kao standardni izlaz. Po definiciji je to korisnikov ekranski terminal. Datoteka 2 je otvorena za pisanje i koristi se za ispisivanje grešaka. Po definiciji je to, takođe ekran terminala.

Skeletni sistem može mijenjati dodjeljene ulazno-izlazna datoteka. To se postiže specijalnim znakovima <, >, << ili >> u liniji naredbe ispred imena datoteke za koju se želi da bude standardni ulaz, izlaz ili izlaz za greške. Pogodno je da izlaz za greške uvijek bude ekran terminala, kako ne bi poruke o greškama bile „sakrivene" u nekoj datoteci.

Jedan od načina preusmjeravanja može se prikazati slijedećim primjerom. Upotrijebićemo se slijedeći oblik naredbe:

naredba << riječ

U konkretnom slučaju za naredbu

sort << kraj

doći će do usmjeravanja standardnih ulaza na **sort** naredbu sve dok se ne unese riječ **kraj**. Tada će doći do sortiranja ulaznog teksta i prikazivanja na ekranskom terminalu računara.

```
$sort << kraj
> avi se mo pričaju o vremenu
> niko ništa ne radi
> kraj
```

i prikazaće se

```
niko ništa ne radi
avi samo pričaju o vremenu
```

Značajan kvalitet UNIX-a u poređenju s nekim drugim operativnim sistemima je mogućnost ulančavanja procesa, odnosno stvaranje kanala ili „cijevi" (pipes) kojima se izlaz iz jednog procesa dovodi na ulaz drugog procesa. Princip je isti kao i kod preusmjerevanja ulaz — izlaz u neku datoteku.

Prilikom poziva naredbe ulančavanja kreira se sakrivane, privremena datoteka zvana *pipe* ne principu fifo (first in first out) reda, koja omogućava procesima da rade paralelno i uz sinhronizaciju sistema, te prenose podatke iz jednog procesa u drugi.

Notacija za povezivanja dva procesa kanalom je vrlo jednostavna. Između dvije naredbe potrebno je ukucati znak „|", što je dovoljno da skeletni sistem pokrene mehanizam ulančavanja procesa

ls — l I wc

Kanalr omogućavaju izvršavanje nekoliko naredbi u nizu sa lijeva na desno. Standardni izlaz svake naredbe postaje standardni ulaz sljedeće naredbe. Korištenjem kanala dobiva se na brzini rada. Kanali su jednosmjerni a sinhronizacija se postiže zaustavljanjem naredne naredbe kada nema ništa za čitanje i zaustavljanjem prethodna naredbe kada je kanal pun.

Jednostavna notacija ulančavanja ima značajan uticaj na programsku metodologiju korsnika skeletnog sistema. Korisneći jednostavnost ulančavanja može se pristupiti kombinovanju postojećih programa, umjesto gradnje

novih programa. Ideja je da se od niza malih programa kombinuju složeniji moduli sa određenim ciljem. Na taj način lakše je definisati, dokumentovati i održavati manje cjeline i povećava se pouzdanost modula izvedenih iz osnovnih programa.

VARIJABLE

Varijabla u skeletnom sistemu predstavlja vrijednost niza. Po slobodnoj definiciji niz je kombinacija jednog ili više alfanumeričkih karaktera ili simbola. Varijabla koje se normalno postavljaju na komandnu liniju nazivaju se parametri. U skeletnom sistemu postoje tri vrste parametara, i to:

- pozicioni parametri,
- parametri ključnih riječi
- specijalni parametri.

Pozicioni parametri se kreiraju kada se pozove procedura skeletnog sistema. Skeletni sistem određuje pozicione parametre na slijedeći način:

$(1) $(2) $(3) . . $(9).

Kako je opšti oblik jednostavne naredbe

naredba arg1 arg2 arg3

tada je vrijednost pozicionih parametara

naredba arg1 arg2 arg3 . . arg9
$(0) $(1) $(2) $(3) $(9).

Na primjer, ako se unese slijedeća naredba

neredba priv1 priv2 priv3

tada će pozicioni parametar **$(1)** imati vrijednost **priv1**. Imenu naredbe procedure uvijek se dodjeljuje **$(0)**.

Pozicioni parametri se često koriste u programima skeletnih sistema. Vrijednosti parametara mogu biti neposredno navedene u trenutku unošenja same naredbe ili ako se naredba nalazi u programskoj datoteci u trenutku njenog aktiviranja.

Takst datoteke skeletnog sistema **Ljeto** je

```
echo Ovo je program $0
echo Ljetnji mjeseci u godini su $1 $2 $3
```

Pozivanjem programa **Ljeto** naredbom

sh Ljeto juni juli avgust

prikazuje se

```
Ovo je program Ljeto
Ljetnji mjeseci u godini su juni juli avgust
```

Parametri ključnih riječi imaju slijedeća značenja:

— HOME — određuje osnovni direktorij korisnika koji je definisan u njegovoj datoteci **profile**.

— PATH — navodi listu direktorija u kojima skeletni sistem traži potrebne naredbe. Može se specificirati nekoliko direktorija u PATH varijabli, ali svaki direktorij zahtjeva određeno vrijeme prije nego se naredba pronađe.

— CDPATH — navodi listu direktorija među kojima naredba od traži potrebni direktorij.

— MAIL — određuje ime datoteke u koju se kopira elektronska pošta naredbom **mail**. Korisnik se obavještava o svakoj novoj poruci koja dospije u navedenu datoteku.

— MAILCHECK — određuje vrijeme u sekundama poslije kojeg skeletni sistem provjerava dolazak nove elektronske pošte u datoteku navedenu parametrima MAIL i MAILPATH. Po definiciji je 600 sekundi.

— MAILPATH — određuje listu datoteka za primanje elektronske pošte. Nazivi datoteka su odvojeni znakom tačka-zarez. Na kraju naziva datoteke moguće je navesti znak % i lakst poruka koja će se pojaviti kada stigne nova pošta.

Tabela 1

Parametar	Značenje
$ parametar	Vrijednost parametara. Umjesto samog naziva parametara može biti * i u tom slučaju $* predstavlja vrijednost svih pozicionih parametara naredbe
$ (parametar)	Vrijednost parametra. Može biti dio drugog izraza. Ako je **$x=Test,** naredba **echo Ovo su $( ) ovi** prikazuje **Ovo su Testovi,**
$ (parametar — riječ)	Vrijednost izraza je vrijednost **$parametar** ako je ona definisana i nije nula, u protivnom je vrijednost koju navodi riječ. Ako je **$x=Test** i **$y=** ' naredbe **echo $(y -Jedan) and $(x -Dva)** prikazuje **Jedan Test;** Vrijednost **x** je **Test**, tako da izraz **$(y -Dva)** dobija vrijednost **$x;** Vrijednost **y** je nula i izraz **$(y -Jedan)** dobija vrijednost **Jedan,**
$(parametar =riječ)	Ako vrijednost parametra nije definisana, parametar dobija vrijednost **riječ;** Ako je **$x=** naredbe **echo $(x =Jedan)** **echo $x** prikazuju **Jedan** **Jedan**
$(parametar? riječ)	Vrijednost izraza je vrijednost **$parametar** ako je ona definisana i nije nula, u protivnom, skeletni sistem prikazuje oba izraza (**$parametar** i **riječ**) i prekide rad. Ako je riječ izostavljena, poruka je „parameter null or not set", Tekst programa skeletnog sistema je **echo $(x?' da li je parametar x definisan') echo Jeste** Aktiviranjem ovog programa ako je **x=Test,** na ekranu se dobija rezultat **Test** da li je parametar **x** definisan **Jeste** Ako je **x=**', program prikazuje rezultat prve echo naredbe **da li je parametar x definisan** i završava rad
$(parametar'+riječ)	Ako je parametar definisan i njegova vrijednost nije nula, vrijednost izraza je riječ u protivnom, vrijednost izraza je nula. Ako je **x=Tri,** rezultat naredbe **echo $(x'+'x je definisan')** je **x je definisan;** Ako je **x=''** nikakva poruka nije prikazana.

— PS1 — znak osnovnog prompta koji se javlja kada skeletni sistem očekuje slijedeću naredbu. Po definiciji je $ ili # za *root*.

— PS2 — znak sekundarnog prompta koji se javlja kada skeletni sistem očekuje završetak naredbe započete u prethodnom redu. Po definiciji je >.

— IFS — određuje znak koji skeletni sistem konsti za razdvajanje njači. Po definiciji to je znak za prazno mjesto, tabulator i novi red.

— SHACCT — naziv datoteke u kojoj skeletni sistem registruje nazive svih aktivnih programa.

— SHELL — određuje da li je program skeletnog sistema u stven skraćene verzije umjesto *sh* aktivira se *rsh*.

Specijalni parametri se automatski definišu, a imaju posebno značenja za rad naredbi skeletnog sistema. Značenje pojedinih parametara je slijedeće:

$$ — je broj trenutno aktivnog procesa skeletnog sistema. Kako je broj procesa jedinstven petocifren broj, on se često konsti za atvaranje jedinstvenog naziva privremenih datoteka. Privremene datoteke su uglavnom neželjene i one postoje dok se izničito na uklone. Međutim, ponekad se pojavi potreba i za jedinstveno imenovanim privremenim datotekama. Slijedeći primjer ilustruje kreiranja privremenih datoteka u direktorijumu koji se konsti samo za ovu namjenu:

```
priv=$HOME/priv/$$
ls>$priv;
```

$! — je broj posljednjeg procesa skeletnog sistema aktiviranog u pozadini. Ovo je takođe petocifren broj.

$# — pokazuje ukupni broj pozicionih parametara dodaljen skeletnom sistemu, ne računajući sam naziv procedure. Varijabla $# daje najviši broj pozicionog parametra. Prema tome **sh x a b c** postavlja $# na 3. Ovaj parametar najčešće se upotrebljava za provjeru prisustva zahtjevanog broja argumenata:

```
if test $# -- lt 2
then echo 'zahtjeva se dva ili više argumenata, exit fi
```

$ — je niz koji se sastoji od naziva izvršnih oznaka, koje su trenutno uključene u skeletni sistem. Varijabla $ — ime vrijednost xv kada se pretražuje izlaz.

$? — je decimalni broj čija vrijednost zavisi od uspjehe posljednje aktivirane naredbe. Ako je naredba njenom izlaznom kodu. Ako je naredbe neuspješna, izlaznom kodu je dodan oktalni broj 0200.

Prije objašnjenja načina upotrebe parametara u naredbama potrebno je definisati pojmove riječ i nulta vrijednost (null)

Riječ se može definisati na dva načina

1) Kao obićan parametar skeletnog sistema,

2) Kao parametar skeletnog sistema izveden poslupkom zamjene nekoliko parametara U izrazu **$(x =$(y))** x je varijabla čija vrijednost (ako nije već ranije definisana) postaje riječ **$(y)** Ako je **y=Test** i x nije definisan, onda je **x=Test.**

Nula vrijednost je definisana slijedećim jednačinama

```
ime =
ime =""
ime = '
```

Za predstavljanje parametara u naredbama skeletnog sistema, koristi se notacija data u tabeli

NAREDBE

Pri pisanju procedura skeletnih sistema, koriste se određene naredbe Većina ovih naredbi konsle se samo za programiranje, dok ostale imaju neprogramersku upotrebu U narednom tekstu će biti prikazene one naredbe koje se na konsle u programiranju

Naredba **cd** se konsli za promjenu tekućeg direktorija Opšti oblik ove naredbe je

cd (arg),

gdje arg specificira novi direktorij

Naredba **exec** omogućava de se naredba specificira se **arg** izvrši odmah, bez kreiranja novog procesa Opšti oblik ove naredbe je

exec (arg . . .),

gdje je **arg** naziv jedne ili više naredbi ili programe koje treba aktivirati Skeletni sistem aktivira navedene naredbe kao novi potproces Kada se naredbe zevrše kontrolu dobija ponovo prvobitni proces skeletnog sistema

Naredba **hash** kada se izvršava unosi se u specijalnu **hash tabelu** Ova tabele čuva Irag naredbi koje su se konsile gdje su bile locirane i koliko pretrazivanja direktorija je obuhvaćeno u locirangu naredbe Kako skeletni sistem pregleda prvo ovu tabelu, vrjeme koje je potrebno za izvršenje naredbe koja je ranije upotrebljavana se skraćuje Tabela **hash** se ponovo inicira kod svakog prijavljivanja konsnika Brisanje labele se vrši naredbom **hash-r**

Da se prikaže sadržaj hash labele potreb no je unijeti **hash**

Primer **hash**

hits	cost	commend
2*	3	/bin/cal
3*	3	/bin/stty
1*	3	/bin/dl
1*	1	calendar
1*	3	/bin/mail
1*	3	/bin/mv
1*	4	/usr/bin/news
1*	3	/bin/mesg
1*	3	/bin/who
7*	4	/usr/bin/vi
4*	2	/usr/ucb/ls
3*	3	/bin/rm

Kolona **hits** pokazuje koliko puta je naredbe pozvana. Kolona **cost** pokazuje broj čvorova koji su pretraženi za pronalaženje naredbe Kolona **command** pokazuje punt put naredbe

Zvijezdica (*) pokazuje da se lokacija naredbe moze promjeniti kada se promjeni radni direktorij i naredba ponovo izvrsi

Naredba **newgrp** mijenja naziv trenutno aktivne grupe Opšti oblik ove naredbe je

newgrp (—) (grupe).

Unošenje naredbe **newgrp** bez argumenta poslavlja originalnu vrijednost grupe Kada se unese (—) uspostavlja se okruženje definisano pri prijavi konsnika

Naredba **pwd** prikazuje puno ime puta tekućeg direktorija. Ova naredba je posebno korisna kada se radni direktorij često mijenja

Naredba **set** omogućava promjenu vrijednosti nekih osobina skeletnih sistema postavljanjem određenih oznaka Najkorisnije oznake i njihovo značenje je

— a — označava nove ili promjenjene varijable za upotrebu sa naredbom **export**,

— f — poništava specijalno znacenje simbola * i ? pri navođenju naziva datoteka

— v — ispisuju se linije koje skeletni sistem cita Naredbe na svakoj liniji unosa se izvršavaju nakon što se te linije ispisuju,

— x — ispisuju se naredbe i njihovi argumenti redosljedom kako se izvršavaju Stvare se trag samo onih naredbi koje se trenutno izvršavaju

Naredba set je naročito korsne za otkrivanje grešake unutar procedura skeletnog sistema Kada se pozove na izvršenje bez argumenata prikazaće vrijednost parametara u okruženju

Primjer **set**

```
HOME=/
IFS=
LOGNAME=root
MAIL=/usr/mail/root
MAILCHECHK=
PATH=./usr/ucb./bin./uer/bin
PS1=#
PS2=>
TERM=vt100
TZ=YUG-1:0YUG
```

Naredba **type** pokazuje kako će navedena naredba biti interpretirana, ako se navede kao ime naredbe Takode pokazuje kompleitnu listu direktorija do naredbe Opšti oblik naredbe je

type (imenaredbe)

Primjer **type export**

Prikazuje *se*

export ie a shell builtin

Naredba **ulimit** ograničava veličinu datoteke koju krera skeletni sistem I njegovi procesi **na n** blokova i ima opšti oblik

ulimit (−f, (n).

Kade se koristi opcija **−f**, ill kada opcija nije specificirane ova naredba nameće ograničenje od n blokova za veličinu deloteke koju skeletni sistem i njegovi potprocesi mogu kreirati Uslovna vrijednost n zavisi od samog računara

Naredba **umask** mijenje prava upotrebe datoteke i ima opšti oblik

umsek (nnn),

gdje je **nnn** oktalni broj koji odreduje prave na upotrebu datoteke Ako je **nnn** izostavljeno lada se prikazuje trenutne vrijednost Naredba **umesk** odreduje prava koja seu uskraćena i obicno je 022, sto znači da samo vlasnik ima pravo na promjenu datoteke

Naredbe **unset** poništava promjenljivih i funkcija i ima opšti oblik

unset (ime),

gdje je **Ime** naziv promjenljiva ili funkcije Ne može se koristiti sa promjenljivim PATH, PS1 PS2 MAILCHECK i IFS, jer se ove promjenlje ve ne mogu ukloniti.

U sledećem broju
Programiranje u skeletnom sistemu

b4browse d4delete itd koje su direktna za mene odgovarajućih naredbi *dBASE-a*

Veličina EXE daloteka napravljenih se *Code Base* bibliotekom zaista može biti mala kod jednostavnih primera – ispod 20 K – ali za role korisnju program moreje uključiti sve elemente biblioteke a tada se dobija datoteka od najma nje 150K.

Pilenje koje je varovatno najosetljivije je koliko *Code Base* pruža, a kokko oduzima. Naime pisanje oplikacije je sigumo malo otežane u odnosu na *Clipper* zbog bitno nizeg nivoa na kome se"radi Ali zbog tog nizeg nivoa pr mer ima daleko veca mogućnosti za specifičnim prepravkama i uopšte više da ulića na rad celog sistema Što se tiče derija performansi problem uopšte nije jednostavan kao sto izgleda Naime se lako mogu podesiti tako da jedan pro deje maksimalno dobre, a drugi maksimal loše razultate Stvar je u tome što veoma mnogo faktora utiče na brzinu raznih operacija Na primer *Code Base* dozvoljava precizno podešavanje memonjskih batera za komunikaciju sa datotekema Ako se podaci učitavaju sekvencijalno veliki bafer ce se pokazati nekoliko puta bolje od malog Medutim eko se vaju male količine rasutih podataka, pri nju" velikog batera Iroši se ogromno vreme svaki slog Slično je i sa *Clipper*-om ako nar ime EMS memonju, *Code Base* to neće primetiti dok se *Clipper*-u značajno poboljšavaju performanse Zato je na priloženom grafikonu za svaki rezultat vršeno više merenja, u raznim uslovima i to lako da bi se uvek dobile maksimalne performanse I programi za testiranje su uvek pravljeni za dve verzije za jednostavan izraz i kompleksan izraz Na primer *Code Base* je pretražio datoteku od 5000 slogova za 12 sekundi po jednostavnom i ali je zato po malo složenijem pretraživao 37 sekundi! Rezultati su dobijeni nak proseka ove dva vremena Merenje je ne relativno skromnoj konfiguraciji (AT ne 12 MHz 1 MB rama pristup disku 28 ms) tako da i to treba uzeti u obzir Korišćeni su *M soft C 6.0 Clipper 87* i *Clipper 5.01*

Indeksiranje daloteka od 5000 slogova najbrže je uradio slan *Clipper Summer 87* – 16 sekundi ali i svi ostali se neznatno razliku ju po vremenu U pretraživanju iste datoteke ubedljivo najbrži je bio *Clipper 5.01* dok je *Code Base* sa NDX indeksima bio cak 8 puta sporiji U pretraživanju po proizvoljnom uslovu (**LOCATE**) *Code Base* je bio veoma spor Istina mora se napomenuti de je to zato sto izrazi za pretraživanje prevode u pseudo kod a ne na mašinski jezik **PACK** datoteke sv de za približno isto vreme (rezultati ne d mu su pomnozeni sa 10) sa napomenom *Clipper 5.01* u slučaju nedostatka memonje ovoj operaciji pokazuje zasta katastrofalne rezultate

Sve u svemu *Code Base* ima veoma ne performanse U sustini ovakva merenja na treba da vam budu osnovni kriterijum de šte nešto konstni Jer u realnom programu se nabrojane operacije skoro nikada ne nalaze u ciklusu, tako da se razlika skoro uvek svodi na nekoliko sekundi (naravno uvek ima izuzetaka) sto je složicale se zenemarljivo. Osnovna prednost *Code Base*-a je pranostvost što se u određenim situacijama može nadoknaditi Kada se uzme u obzir i moć *C* jezika *Code Base* je svakako proizvod na koji treba računati

Zahvaljujemo se Institutu za računarsku tehniku i telekomunikacije iz Beograda na softveru koji nam je stavljen na raspolaganje za potrebe ovog testa

KO JE ZALUTAO U CAD

Poštovani gospodine Regesek

Kao dugogodišnji čitalac i povremeni saradnik, želeo bih da pohvalim boljí kvalitet štampe i novi izgled časopisa. Na visoki profesionalni nivo objavljenih priloga čitalačka publika je već navikla.

Upravo iz ovih razloga bio sam prilično iznenađen površnim i neodgovornim pnkazom *Nth Drive* dnjvera u septembarskom broju autora gospodina Novakovića.

Razočaran sam totalnim neposnavanjem čak i osnovnih funkcija proizvoda *Nth Drive* softvera koje je autor spojio. Zabnnjava autorov nedostatak sposobnosti ili želje da se upozna proizvod a zatim isti evaluira. Neobično je to što pnstup gospodina Novakovica odudara od uobičajene i prihvaćena metodologije evaluacije kako hardverskih tako i softverskih proizvoda koja je jasno uočljiva kod ostalih autora iz ise kuće. Upravo zahvaljujući njima i takvom načinu rada. Računari posedujú ovoliko brojnu publiku.

Pored pnkazane stručnosti iz oblasti tehnika i novinarstva, gospodin Novaković se predstavlja i kao poznavalac tržišta. On pomenutom softverskom proizvodu određuje vrednost od 50 dolara. Činjenica da se *Nth Drive* vrlo uspesno prodaja za 595 dolara u zemlji koja je izmislila tržište računara i softvei može na samo da stavi pod sumnju autorova procene već i dobre namere. Ovo pogotovo zbog toga što je u više navrata odbijana dobronamerno ponudena pomoć i saradnja od strane stručnjaka iz ABV Graphics-a ili pak, korisnika *Nth Drive*-a.

U prilogu vam dostavljam kopiju pisma upućenog gospodinu Novakovicu sa konkretnim primedbama

Mr S. Radić

Poštovani gospodine Novakeviću

Želimo da iskažemo naše zadovoljstvo što ste odvojili prostor za prezentiranje Nth Graphics ovog softverskog proizvoda – *Nth Drive*

Časopis „Računari" pokriva vrlo široku oblast personalnog računarstva i uloliko nam je dnža sio sle se posvetili i profesionalnim proizvodima namenjenim užem krugu korisnika. Svesni smo da je Vaše opterećenje kao stručnog saradnika časopisa vrlo veliko kako po količini tako i po raznovrsnosti proizvoda koja testnate ocenjujete i komantanšete i da se vrlo često nalazite u vremenskom lesnacu

Zbog navedan h razloga još jednom Vam nudimo našu saradnju i pomoc kod avaliacije Nth Graphics-ovih proizvoda kako bi se što lakše i brze upoznali sa njihovim karakteristikama i performansama

U vezi vaša evaluacija *Nth Drive* softvera imamo nakoliko dobronamernih komentara. Smatramo da su inženjeri projektanti erhitekte i ostali koj koriste *AutoCAD* u svom profesionalnom radu, pored jlanen lamih podataka koje ste naveli, ostal uskiaceni i za niz bitnih funkcija i informacija o njima.

Nth Drive pored oslalog obezbeđuje i

Template mogucnost podmatanja slajda pod le kući crtež što pored očigladnih dnugih primena ima za rezultat dramatično skraćivanja „select" vremena kod slozenih crteža

Control Panel kontrolni panel pomoću kojeg je moguce jednim potezom misa lisiat direktorijume kako na svom tako i na PC-jima u mrezi učitavati lisp rutine blokove slajdove i DWG datoteke- uključivali i isključivali pojedina boja kako na crtezima tako i na slajdovima kao i razne veličina i tipove slova kako za bočne tako i za spuštajuce menije što je vrlo korisno a katkad i neophodno pn radu se n skim odnosno vrlo visokim rezolucijama. Prilagođavanje *AutoCAD* menija i maski prema vlastitim potrebama i ukusu

Nth HELP stalno resploživa pomoć za Nth funkcije

HYDRA jednomlani softvei za manipulaciju vremenskog učitavanja i senčanja *AutoCAD*-a i *HYDRE* čime se obezbeđuja istovremeno korišćenje ova dva programa. Mogucnosi tada sa dva monitora u kom slučaju se na jednom monitoru radi sa **HYDROM** a na drugom se *AutoCAD*-om

Nth Render softver koji omogućava korišćenje softvarskih paketa *3D STUDIO* i *AutoShade* na obič-

nim i boljim VGA karticama. al sa rezolucijama od 1280 x 1024 i sa emulacijom 16 7 miliona boja. Uključen u cenu *Nth Drive*-a

Tri knjige (*Nth Drive*, HYDRA i Nth Render) sa uputstvima za nstalaciju i korišcenje kao i sa poiuka ma o greškama sve se primenma. Ukupno oko 300 strana

Dve video kasete kao pomoć za brzo i afikasno upoznavanje svih funkcija *Nth Drive*-a i *HYDRE*

Drajveri za najrazličnije VGA kartice za rad u pre ko 225 rezima

Obezbedena **jednogodišnje besplatne podrška** i obrzavanje softvera

Uzimajući u obzir kvalitet i količinu softvera kao i pratecu dokumentaciju i podršku, može se reći da Nth DRIVE vredi $ 595 i da zaslužuje hardversku zaštitu

Na kraju treba reći da je Nth Drive više od softver skog drajvera. Nth Drive je proizvod

Naravno, sve rečeno ima značaja kod profesionalca a amaiten i povremeni posetioci CAD a i ne treba da za svoj softver odvajaju vise od 50 dolara kao ni da troše svoje dragoceno vrama

Mr. S. Radić
Tehnički direktor

P.S. ABV Graphics je ekskluzivni distnbutar firme Nth Graphics

Poštovani g. Radiću.

Zahvaljujem na primedbama i sugestijama u vezi sa pnkazom Nth Drive proizvoda koji vi prodajete. Reakcije makar i negativne, uvek tad je prinam nego potpunu ravnodušnost prema lirže šiu i štampi

Vaše glavne primedbe se odnose na moj stav prema ovom proizvodu. Evo mojih argumenata:

1) *Display List* drajvan za *AutoCAD 11* ovog reda cene (> 600 USD + dežbine) za SuperVGA i slabije 8514A karte namaju mnogo smisla na zeista profesionalnim sistemima, na kojima vi insistirate. Stvarne dobitke u brzini *Nth Drive* donosi tek na sistem ma rada 33 ili 50 MHz 486 sa 8 i više MB RAM – tek tada je



procesor dovoljno brz da se zaiste dobiju željena ubrzanja a i RAM je dovoljno veliki da. bar am kod manjih modela. bh pola ili 1 MB odvojenih za ekransku listu ne naletaju osnovni crtež da se stran č na disk čime bi bila poništena konsi od ubrzanja koja bez dvoumljenje *Nth Drive* zaista donosi. Kod takvih zaista profesionalnih i ne baš jevtnih s stema mnogo isplativija stvar je dati još par stoti na dolara v še i kupiti na primer Hercules Graphics Station Card – 2 sa svojim 60 Mhz 34010 i 1 MB brzog VRAM i 2 MB posebnog DRAM za ekranske i ste koja prilom izvrsava grafički procesor a na kao kod *Nth Drive* glavn procesor. Uz to HGS Card – 2 raspolaže i celim nizom grafičkih naći na sa 32768 i sv h 16 7 mi ona boja neposredno dostupnih i nalazno potpunom IBM VGA kompatibilnošću. Pn tome je jedan od načina (512 × 480 × 24) naposredno kompatibilan sa TARGA karticama koje inace imaju starju i slabju verziju procesora koji je na HGS Card – 2. Poslednje ali i najvažnije je što se uz ovu zaista izvanredn kartucu pored veznika za prekidno sav poznatiji softvai dobijaju i izvisni protected-mode *Display List* drajveri za *AutoCAD 11* se bogatstvom opcija veoma bliski m *Nth Drive* i *AutoShade 2* i *3D Studio* renderingom na svim rezolucijama kartice

2) Jedna od grafičkih kartica koje konsim je i Diamond Speedstar VGA Plus (informacije MICRO AIR tel 453 3521). Kartica sa ET4000 čipom 1024 × 768 na 70 Hz sa 256 boja i opcionim 800 × 600 sa 32768 boja istovremeno dostupnih, pored prilično velike brzine rada dobija se i sa celom gomilom izvanrednog softvera. Pored pouzdan h drajvera za sav mogući i namogući softver dobijaju su i izvanredni grafički program *WinRIX* i nalazno *Panacea protected mode display*-ksi drajveri za *AutoCAD 11*. Ovaj drajveri posed brzina omogućava i rendering u *AutoShade 2* i *3D Studio* paketima na svim rezolucijama kartice kao i vaš NthRender. O kvalitetu i pouzdanosti imana Panacea je dovoljno reci da su IBM i Hercules dva mena koja su ulamlajila PC grafiku zadužila Panaceu za izradu DLD *AutoCAD* drajvera za svoje IBM XGA odnosno HGS Caid kartice. Za 50 USD vlasn ci ove kartice mogu dokupiti i Deluxe verziju ovog drajvera i ne laj način imali praktično sve mogućnosti *Nth Drive* (Birdseye, Spyglass, Template) i sve to (hardvai + softver) za cenu koja je oko dva puta nize od cene samog *Nth Drive*

3) U pismu je navedena i cela gomila dodatakal stvan (HYDRA dva video kasete itd) koje po vasim rečima pripadaju paketu. Na redakcijski sto su na mali rok bila slavljena dva kutija vašeg softvera. One su bile potpuno nezavisne i pošto su imale i nezavisne hardverske zaštite, log čan zaključak je bio da su to dva nezavisna softverska paketa. Nikakvih video kaseta i sličnih stvari tu nije bilo a Računari testiraju ono što dobiju. Usled uslova vaše ponude za testiranja viamana za poseban pnkaz HYDRAview AC jed nostavno nije bilo. Od sada ponuda testiranja softvera sa takvim klauzulama uopšte nece biti razmatrane.

Moram priznati da je moje mozda neopravdano razočaranje se *Nth Drive* bilo u velkoj meri izazvano i lom brzinom rada koju su uslovi testiranja diktirali. U avakom slučaju, *Nth Drive* na boljoj 8514A karticl kao što je inace pieskupa NthEngine/150 bi zajedno sa HYDRAview/AC vizualizatorom bio zanimljiva tema za komparativni test se pai sličnih hardversko-softverskih paketa. U svakom slucaju nedostatka želja za testiranje nije bilo a o sposobnosti neka čitaoci donesu svoj sud.

Ipak, u potpunosti odbacujem sve vaše optuzbe o nekakvom amaterizmu i nekakvim „povremenom poselama" CAD i grafici. Čitaoci „Računara" upoceni u ovu problamatiku mogu primetiti da smo se i redakcija i ja od rođenja ove rubnike trudili da tekstovi budu ne samo detaljni objektivni i maksimalno konsni nego i gde god je bilo moguće propraceni originalnim modelima koje sam za te prilike kreirao CAD i high-end grafika su i inače jedno od mojih osnovnih životnih usmerenja u koje sam ulozio mnogo godina rada. Ako objektivan stav nekome ne odgovara to je već drugo pesma. Tome b umesto o mom amatenzmu i povremenim poselama CAD" bilo pametnije da poraizmisli o sopstvenoj poslovnoj politici povremenog poseboca dzepova kupaca.

Nebojša Novakovic

LAKI UVOD U ADU

U ranim sedamdesetim godinama Ministarstvo odbrane SAD-a (DoD) je raspisalo konkurs za novi programski jezik, pre svega zbog visokih troškova pri projektovanju softvera, a zatim i zbog standardizacije. Pobedio je jezik sa radnim nazivom *Green* firme CII Honeywell Bull. Posle odredenih izmena, jezik je dobio ime ADA.

Siniša Ćurčin

Ada je bila ćerka Lorde Bajrona i smatra se prvim programerom. Korišćenjem ade efikasnosti u projektovanju softvera se prame nekim izvonma povecala želimi pula. Ne namjeravam da tvrdim de je ada najbolji jezik za programiranje, ali u svakom slucaju ima mnogo prednosti u odnosu na ostale programske jezike. čak i ako zanemarimo ugled i autoritet (i dolare) ustanove (DoD) koja je adu prihvatila kao svoj standard.

Zbog cega su se u DoD u odlučili za adu? U ovom članku ce biti prikazane samo neke karakteristike ade koje nemaju ostali programski jezici i koje su doprinele de lej programski jezik izbori lako visok status.

TIPOVI PODATAKA

Oni koji koriste fortran znaju da izrazi

1/2 i 1./2

nikako nisu isto – u prvom slučaju rezultat je 0, a u drugom 0.5. Zamislite modul koji se sastoji samo od formula i koji daje pogresan rezultat, a nedostaje samo jedna tačka. Zaista je potrebno mnogo vremena i slpljenja da bi se pronašla takva greska.

Ada zahteva strogu tipizaciju promenljivih. Ako deklarišete promenljivu A kao tip **pritisak** a promenljivu B kao **temperatura**, prevodilac vam neće dozvoliti da saberete te dve promenljive iako su oba podatka realni brojevi – osim ako to izričito ne naglasite. Ovaj primer "pokazuje i jednu drugu osobinu ade: možete definisati onoliko tipova koliko želite:

```
type pritisak is new float,
type temperatura is new float,
```

a zatim i deklarisati promenljive koje su tog tipa:

```
A: pritisak,
B: temperatura;
```

Treba uočiti razliku izmadu deklaracije promenljivih i definicije tipova. Fortran, paskal, C i njima slični jezici nemaju mogućnost definisanja novih tipove. Naravno, ne uzimajuci u obzir podatke tipa **record** u Paskalu recimo, jer svi takvi podaci imaju jeden isti tip **record**. Ovda se radi u stvari o suštinski novom pristupu.

Pored „standardnih" tipove promenljivih u adi posto je dva tipa realnih brojeva – sa fiksnim i pokretnim zarezom:

```
type VISINA is delta 0.1 range 0.0..10000.0;
-- fiksni zaraz
type BRZINA is digits 8 range 0.0..120.0;
-- pokretni zarez
```

Iz primera se može videti da se svakom tipu moze zadati odredene oblasti definisanosti i broj znacajnih cifara.

Pokušajte da definišete matricu u fortranu sa indeksima kao u ovom primeru:

```
type matrica is array (-10..10,-20..20),
A, B, C: matrica;
```

i nećete uspeti jer je to nemoguce. Ako su vam zaista potrebni takvi indeksi, morscete de iskomplikujete druge stvari.

Dijapazon indeksa zeista može da bude bilo koji:

```
type RADNI_DANI is (ponedeljak, utorak, sreda,
cetvrtak, petak),
type ZARADA is array(RADNI_DANI),
-- ovo je vektor
```

Ili da se i ne zada

```
type VEKTOR is array(INTEGER range <>) of
FLOAT,
```

Tu nije kraj. Da bi se pomnožile matrice A i B, morali bisle da napišete nekoliko linija koda, a u adi je dovoljna jedna:

```
C := A*B
```

Inicijalizacije vrednosti takode nije problem. Ako svim elementima vektora A treba da dodelite vrednost 1, nije vam potrebna petlja:

```
A:VEKTOR(1..100) := ( others => 1.0),
```

PROGRAMSKE PETLJE

U adi postoje više vrsta petlji kao i u drugim jezicima. Novost je da svaka petlja može da ima ime i da se to ime može koristiti u instrukcijama za izlazak iz petlje (primer na listingu 1). Takode specifična je i **for** petlja:

```
for I in 1..100 loop

end loop;
```

Varijabla I se ne mora eksplicitno deklarisati (izgleda da je nekome od projektanala ade uvek nedostajala varijabla za for petlju, pa je taj problem rešio na ovaj nacin).

Uporedite kod u adi (listing 2a) sa kodom u C-u (listing 2b).

U oba primera se radi ista stvar, ali... Možde ce neko reci da je bez obzira na čiljivost koda u adi C ipak jezik za „prave" programere, ali ce i „pravi" programer morati da priznaju de ukoliko jednog dana narodni poslanici odluče da radimo i subotom, kod u adi treba ispraviti na jednom mestu, a u C-u na tri masta (od deset linija koda). Ako se poveca broj linija koda normalno je i da se poveća broj ispravki, ako znate GDE treba da izvršite ispravke.

OBRADA ISKLJUČENJA

Zamislite da ste napisali softver za prenos podataka sa jednog racunara na drugi i posle 24 sata rada dode do greške koja je prouzrokovana delenjem sa nulom. Naravno, operativni sistem ispiše odgovarajuce poruke i prekine se izvršavanjem vaseg programa. Ne preostaje vam ništa drugo nego da ponovo startujete program i molite se da sledeći put imate više srece. Pilot u avionu u takvom slučaju sigurno nece imati priliku da ponovo pokuša. Medutim, ada i ovog puta pruza odgovarajuce resenje. U svakom modulu možete definisati postupak u slucaju da dode do grešake za vreme izvrsavanje programa koriščenjem instrukcije **exception** (listing 3).

Kada dode do greske prekida se izvrsenje programa i pralazi na izvršenje instrukcija koje su predvidene za taj slucaj.

PROCEDURE I FUNKCIJE

Ada ima i procedure i funkcije kao i ostali jezici. Ali ostali jezici nemaju ono što ima ada. Pogledajmo primer na slici 4, koji rešava problem sabiranja matrica bez obzira na dimenzije i definisan operator sabiranja. Zaista, samo u adi mozete pisati za matrice A, B i C:

```
C := A+B;
```

U C-u postoji nekoliko procedura za konverziju broja u string sa različitim imenima u zavisnosti od tipa parametra: **itoa**, **ftoa**, **ltoa**. U svi svim takvim procedurama mozete dati isto ime, iako su one u stvari različite. Prevodilac ce na osnovu tipa parametara iskonstruisati odgovarajuću proceduru (jer one zaista suštinski rade istu stvar).

Još jedna velika specifičnost – prilikom poziva procedure nije obavezno da se navedu svi parametri. Procedura:

```
procedure PUT(C: in CHARACTER := 'B'; in BOOLEAN := RED),
```

može se pozvati na sledeće načine:

```
PUT,
PUT(C => 'e'),
PUT(BOJA => BLUE, C=> 'S'),
PUT( 'S',GREEN)
```

PROGRAMSKI MODULI

Paketi, potprogrami, zadaci i generički moduli predstavljaju celini vida programskih modula od kojih se sastoji program u adi.

Paket se sastoji iz dva dela: specifikacije paketa i tela paketa. Za korsnika je vidljivo samo ono što je deklarisano u specifikaciji paketa:

```
package STACK is
        procedure PUSH(X: REAL);
        function POP return REAL;
end;
```

a način na koji su implementirani potprogrami PUSH i POP za korsnika nije bitan. Što je još važnije, POP i PUSH se mogu koristiti i bez tela paketa i bez obzira na to kako su implementirani. Telo paketa se ne koristi prilikom prevođenja vec samo prilikom linkovanja (listing 5).

U telo paketa su implementirani POP i PUSH.

Ove specifičnosti ade ja naročito pogodna za timski rad. Kada se jednom definišu procedure i njihovi parametri, svi clanovi tima mogu da rade u isto vreme nezavisno jer više ništa ne moraju da znaju o posiu kojeg njihove kolege rade.

Zadaci (task) su procesi koji se izvrsavaju paralelno. Mnoge probleme je daleko lakše rešiti ako se koristi takav prislup rešavanju.

Posmatrajmo školski primer koji se veoma elegantno moze resiti primenom leskova. Problem je predlozio E. W. Dijkstra:

Pet filozofa provode vreme razmisljajuci i jeduci makaronе. Oni sede za okrugtim stolom na kojem se nalaze pet tanjira i pet viljuska: izmedu svaka dva tanjira po jedna. Da bi jeo, filozof mora da ima dve viljuske i kada ih ne koristi mora da ih spusti na sto.

Primetite da filozofi ne mogu da jedu svi u isto vreme. Njihove radnje se mogu modelirati pomocu paralelnih procesa na način dat na listingu 6.

Pogledajmo proceduru SWAP(A, B) koja zamenjuje vrednosti promenljivama A i B. U zavisnosti od njihovog tipa morali bismo da pisemo uvek novu proceduru. Ili u najboljem slučaju da prepravljamo stare. Naravno u adi za sve postoji rešenje. Jednostavno se napravi „šablon":

```
generic
        type ELEMENT is private;
procedure SWAP(X,Y: in out ELEMENT);
```

Listing 1

```
...
...
SPOLJNA:
loop
   ...
   SREDNJA:
   loop
      ...
      UNUTRASNJA:
      loop
         ...
         when e=1 exit SPOLJNA,
         when a=2 exit SREDNJA;
         when a=3 exit UNUTRASNJA;
      end loop UNUTRASNJA;
   end loop SREDNJA;
end loop SPOLJNA;
```

Imenovane programske petlje

Listing 2a

```
...
type RADNI_DANI is (ponedeljak,utorak,sreda,etvrtak,petak);
type ZARADA is array(RADNI_DANI);  -- ovo je vektor
...
for I in RADNI_DANI loop
   PUT_LINE(ZARADA(I));
end loop;
...
```

Primer *for* petlji u ad...

Listing 2b

```
...
#define PONEDELJAK 0
#define UTOKAK 1
#define SREDA 2
#define CETVRTAK 3
#define PETAK 4
...
float zarada[5];
int i;
...
for (i=PONEDELJAK; i<=PETAK;++i) {
     printf("%f\n",zarada[i]);
}
...
```

...i na C-u

Listing 3

```
begin
   ...
   ...
exception
   when PARITY_ERROR => ...
                        ( postupak1 ),
   when OVERFLOW     => ...
                        ( postupak2 ).
   ...
end;
```

Obrada iskljucenja instrukcijom *exception*

Listing 4

```
type MATRICA is array
   (INTEGER range <>, INTEGER range <>) of FLOAT;
...
function "+"(X,Y: MATRICA) return MATRICA is
   SUMA: MATRICA (X'FIRST..X'LAST,X'FIRST(2)..X'LAST(2));
...
begin
   for I in X'RANGE(1) loop
      for J in X'RANGE(2) loop
         SUMA(I,J):=X(I,J)+Y(I,J);
      end loop;
   end loop;
   return SUMA;
end "+";
```

Sabiranje matrica

Sada procedura SWAP izgleda ovako

```
procedure SWAP(X,Y: in out ELEMENT) is
   T: ELEMENT;
begin
   T:=X;
   X:=Y;
   Y:=T;
end SWAP;
```

Listing 5

```
package body STACK is
   MAX: constant:=100;
   S: array(1..MAX) of REAL;
   PTR: INTEGER range 0..MAX;

   procedure PUSH(X: REAL) is
   begin
      PTR:=PTR+1;
      S(PTR):=X;
   end PUSH;

   function POP return REAL is
   begin
      PTR:=PTR-1;
      return S(PTR+1);
   end POP;
begin
   PTR:=0;
end STACK;
```

Programski paket

Listing 6

```
procedure DINNING is
   subtype ID is INTEGER range 1..5;

   task type PHILOSOPHER is
      entry GET_ID(J:in ID);
   end PHILOSOPHER;

   task type FORK is
      entry PICK_UP;
      entry PUT_DOWN;
   end FORK;

   F: array(ID) of FORK;
   P: array(ID) of PHILOSOPHER;

   task body FORK is
   begin
      loop
         select
            accept PICK_UP;
            accept PUT_DOWN;
         or
            terminate;
         end select;
      end loop;
   end FORK;

   task body PHILOSOPHER is
      I: ID;
      LIFE_LIMIT: constant:=100_000;
      TIMES_EATEN:INTEGER:=0;
      LEFT,RIGHT:ID;
   begin
      accept GET_ID(J: ID) do
         I:=J;
      end GET_ID;
      LEFT:=I;
      RIGHT:=I mod 5 + 1;
      while TIMES_EATEN /= LIFE_LIMIT loop
         P(RIGHT).PICK_UP;
         P(LEFT).PICK_UP;
         P(LEFT).PUT_DOWN;
         P(RIGHT).PUT_DOWN;
         TIMES_EATEN:=TIMES_EATEN+1;
      end loop;
   end PHILOSOPHER;
begin
   for K in ID loop
      P(K).GET_ID(K);
   end loop;
end DINNING;
```

Paralelna obrada različitih procesa

Ako nam je potrebna procedura za zamenu vrednosti dveju matrica jednostavno je deklarišemo

procedure MATRIX_SWAP is new SWAP(MATRIX)

i nije potrebno pisati novu proceduru. Na ist način se dobija i procedura za zamenu vrednosti bilo koje dve variable drugog tipa.

TO NIJE SVE

Zaista ovim nisu iscrpljene sve prednosti ade u odnosu na druge jezike. Samo nabrajanje svih karakteristika bi zauzelo prostora koliko i ovaj članak — o tome nekom drugom prilikom kao i o prednostima korišćenja ade pri rešavanju pojedinih problema.

Zbog cega se, ipak, ada još uvek ne koristi kod nas onoliko koliko zaslužuje? Najvažniji razlozi su: skupi kompajleri (*Ada Vantage* je oko pet puta skuplji od *Turbo C*-a), nedostatak literature i nedostatak biblioteka sa programima. Stanje se, naravno popravlja, pa se ne tržištu mogu naći biblioteke za adu (na primer grafička biblioteka HALO88), iz ade se može pristupiti ORACLE-ovoj bazi podataka itd. Očekuje se da će ovakve podrške biti sve više.

SKUPLJANJE DATOTEKA

Na većini BBS-ova širom sveta datoteke aa javnim softverom predstavljaju najvažniju stvar koja se na njima može naći. I na velikim računarskim mrežama, kao što je Bitnet, one su takode važan segment. Dovoljno je imati pristup nekom od *mainframe* računara (kod nas uglavnom VAX-ovima povezanih u DECNet) koji su priključeni na BITNET / EARN i postaće vam dostupni gigabajti javnog softvera.

Jovan Bulajić

U ovom članku ćemo se baviti jednim od načina da dodete do tih programa, a ze druge načine kako korisno da upotrebite svoj (ili tuđi) račun na nekom od računara vezanih na DEC-Net! konsultujte seriju članaka Dajana Ristanovica o elektronskoj pošti. Pročitajte te tekstove i ako ste početnik na VAX-u pošto se ovde nećemo mnogo osvrtati na objašnjavanje osnovnih komandi i operacija.

ŠTA JE TRICKLE?

Trickle je file server na Bitnetu, ogromnoj računarskoj mreži koje povezuje hiljade velikih računara i opslužuje nama nepoznat (citaj: ogroman) broj korisnika širom sveta. *Trickle* je praktično server koji omogućava vezu između korisnika u Evropi (povezanih u EARN – European Academic Research Network) i SIMTEL20 – ogromne arhive javnog (*public domain* i *shareware*) softvera koja se nalazi negde u USA (White Sands Missile Range, New Mexico). Taj arhivi (koja se može nalazi na računaru koji drži američka armija, ali služi isključivo u civilne svrhe) se može osim preko *Trickle*-a pristupiti i na još neke načine. Neki od njih mozda poznajete: ime DDC – Detroit Download Center. To je jedan od najvećih američkih BBS-ova. Ono što se na njemu može naći predstavlja prosečano slanje sa SIMTEL20. Takode se može pristupiti i preko Bitnet FTP servera (u koma ce možda biti reči neki drugi put).

Datoteke na *Trickle*-u su podeljene po direktorijumima, a osnovni su MISC, MSDOS, SIGM, PC-BLUE, CPM, MSDOS2, SUMEX-AIM, EXPO-MIT, UUNET, WUARCHIVE i FUNIC-FI. Mi ćemo se ovoga puta uglavnom baviti za većinu od vas najzanimljivijim direktorijumom – MSDOS koji (logično) sadrži programe za vaš PC. U ostalim direktorijumima je takode ogroman izbor softvera za druge računare, na primer za UNIX sistema (sa velikim brojem programa dalih u izvornom kodu) CPM „mekintoš" NEXT i SUN.

MSDOS direktorijum (katalog) je podeljen na poddirektorijume, kojih trenutno ima čak 171 (kompletan spisak možete videti na slici 1). Tamo se sada može naci oko 6500 programa ukupne dužine oko 400 MB. Svi programi su naravno arhivirani uglavnom PKZIP arhiverom.

OSNOVNE KOMANDE

Red sa *Trickle* serverom je vrlo jednostavan (narocito kada se na njega naviknete) a za to vam neće trebati mnogo vremena). Potrebno je da pošaljete poštu (mail) na adresu *Trickle* servera i da u poruku stavite odredene komande. Poruku saljete na email adresu vama najbližeg (od desetkoliko ih trenutno postoji) lokalnog čvora *Trickle*-a. U našem slučaju, taj čvor se nalazi u Becu, ne email adresi trickle@awiwuw11.bitnet. Najzgodnije ce vam biti ako u ALIAS.COM datoteci definišete sinonim

$ ALIAS trickle „EARN.""TRICKLE@AWIWUW11.BITNET""

pa onda pri slanju poruke (MAIL> SEND) za adresu možete otkucati samo

"trickle"

Komandi koje šaljete *Trickle*-u nema mnogo. Za početak vam preporučujemo da zatrazite dejaljno uputstvo za upotrebu. Iako što cete poslati poruku koja se sastoji iz jednog jedinog reda

/help

Posle izvesnog vremena (ne baš odmah ali u roku od nekoliko sati ukoliko mreža nije zakrčena) dobicete poruku (mail) od *Trickle* servera sa traženim upustvom (dugačko dvadesetak K).

Dva osnovne komande su **/pddir** i **/pdget**. Sa **/pddir** dobijate spisak datoteka (ekvivalentno dir u DOS-u), a sa **/pdget** dajete nalog *Trickle*-u da vam pošalje željenu datoteku. Sintaksa komandi je sledeca (dir=direktorijum)

/pddir <dir.poddir>
/pdget <dir.poddir>datoteka (opcije

Ukoliko kod **/pddir** na navedete nikakav argument (tj. pošaljete samo **/pddir**) dobićete spisak osnovnih direktorijuma. Se

/pddir <msdos>

dobićete spisak poddirektorijume u MSDOS (slika 1) I konečno sa

/pddir <msdos c>

dobićete spisak svih datoteke u direktorijumu **<msdos.c>** (slično kao kada biste deli dir \msdos\c u DOS-u). Sa

/pdget <direktorijum>datoteka (opcije

nalažete *Trickle*-u da vam posalje datoteku koju želite. Na primer

/pdgal <msdos qedit>qedit215.zip

ce vam poslati datoteku qedit215.zip iz direktorijuma msdos.qedit. Možete navesti nekoliko opcija npr. (**SF** od Sand File) saopštave file servaru da datoteku šalje u binarnom (NETDATA) obliku (konsteći SENDFILE za transfer), dok (**UUE** odrediuje da se datoteka pošalje UU kodirana (šta je UU kodiranje biće reči kasnije). Sa (**MAIL** naredujete da vam se datoteka šalje u obliku poruka (MAIL-a). UU kodirana i podeljena u više delova. Naravno (**MAIL** i (**SF** nikako nemojte koristiti zajedno, jer za to nemate nikakve potreba sem ukoliko ne želite da studite server (u dokumentaciji sa spominje da se server u ovakvim slučajevima ponaša „nepredvidivo").

Verne ce najkonsnija bili opcija (**SF**. Ukoliko je na navedele duže datoteke *Trickle* ce podeliti na nekoliko delova i poslati u nekom njegovom formatu (BITSEND). Takve datoteke cela moreti da spojite programom BITRCV ali pošto nismo uspeli da ga nademo naviknli smo se da koristimo isključivo (**SF** opciju. Mana je što lako poslana datoleke putuju nešto sporije, a što bi se moglo desiti da neke duže datoteka uopšte i ne stignu do vas. Ako se desi da *Trickle* neca da vam pošalje neku dužu datoteku, a nemate gorespomenuti BITRCV preoslaje vam jedino mučenje sa (**MAIL**.

U jednoj poruci koju šaljete *Trickle* sarvaru može biti više raznih komandi, svaka u novom redu. Pogledajte sliku 2 gde je dat primer jedne poruke.

Kad pošaljete poruku sa neredbama *Trickle*-u, preostaje vam da čekate rezultate njihovog izvršavanja. Server vam prvo pošalje poruku sa odgovorima na vaše komande gde vam daje rezultate izvršavanja vaših **/help** i **/pddir** komandi dok za **/pdget** daje obavestenje da će vam daloleka biti poslala. Ukoliko ste tražili neku neposlojeću datoleku (ili direktorijum) bićete obavešteni o grešci. Ta greške je obično rezultat vašeg slabog kucanja, ali sa ponekad dešava da sle zatražili neki program baš kad je stigla njegova nova verzija pa je promenjeno i ime arhive. To nam se desilo jednom sa scanv78.zip i scanv80.zip pa smo posle dosla čuđenja (kako je nema kad je u spisku?") naučili za ubuduće da ne treba odmah dizati paniku nego probati **/pddir** jos jednom.

Na poruku-odgovor *Trickle*-a obično ne treba čekati dugo – stigne u roku od jednog dana (sem ako se nija našao neki nadležni pamelnjakovic pe pnvremeno, na primer preko vikenda, zatvono vezu sa Bitnetom).

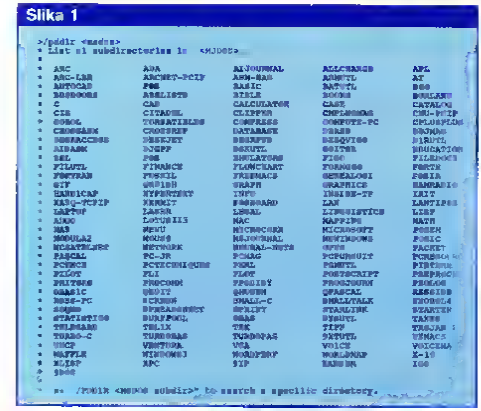
Na same datoleke (posle potvrde servera da će biti poslate) cele se načekati melo više – nema se desavalo da čekamo i po deset dana. Ne uzdujute se mnogo zbog toga – obično cela imali srače da za dva tri dana dobijete ono što sle tražili. Ponekad vam se može posrecil da daloleka dobijete u roku od samo nekoliko sati čak i pre nego što stigne potvrda da ce biti poslate. Najtraženije datoleke *Trickle* drži u nekom svom lokalnom kes direktorijumu i šalje ih odmah, dok se ostale datoleke prvo moraju prebaciti sa servera koji ih drži (sve datolalke na *Trickle*-u su raspoređene ne deset lokalnih servera) a tek onda prosledriti vama.

PRIMANJE DATOTEKA

Zavisno od vaše sreće tj. privilegija koje imate na lokalnom VAX-u pristigle daloleke možete primiti na dva načine. Ukoliko imate omogucen pnstup RECEIVE sistemu (programu) tj. ukoliko VAX na kome radite koristi JNET mrežni softver, možete se zadovoljno osmehnuti i preskočiti redove koje opisuju muke (koje bas i nisu neka muke, ali je postupak nesto kompliikovanij i sporiji) onih koji nemaju tu privilegiju.

Ukoliko niste taj srečnik koji ima RECEIVE daloleke cele dobijati kao MAIL podeljene u više delova i UU kodirane. Morale prenesi sve poruke na svoj PC spojiti njihov sadržaj u jednu deloleku, obično ekslenzije UUE i konvertovati je u normalan oblik programom za UU dekodiranje (možete ga naći i na Sezamu datoteka \com\uuexe413.zip)

Slika 1



Spisak poddirektorijuma u direktorijumu <MSDOS> koji se dobija komandom /pddir <msdos>. Za prlatup nekome od njih traba koristiti /PDDiR MSDOS subdir

Slika 2



Primer komandne poruke

Pošto se verovatno prvi put susrecele sa pojmom UU kodiranja / dekodiranja red je da to ukratko i objasnimo UU kodiranje je postupak kojim se normalni (8-bitni) ASCII karakteri prelvaraju u standardni karakter set (ASCII kodovi od 32 do 95 dakle bez malih slova), koji se onda može slati po svim mogucim (pa i onim davno zastarelim) računarskim mrežama. Osnovni postupak je razbijanje grupa od tri 8-bitna karaktera (ukupno 24 bita) u četiri 6-bitna karaktera na koje se ooda dodaje još 32 (space) da bi se dobili karakteri kodova u rasponu od 32 do 95

UU kodirana datoteka je (očigledno) za svoju četvrtinu duža od originala

Ako imate omogućen RECEIVE startujte ga (otkucajte RECEIVE) i dobićala spisak prispelih daloteka nešto pcpul onoga na slici 3 Sada su vam na raspolaganju dva postupka. Opisacemo prvo komplikovaniji (dal u *Trickle* dokumentaciji) – otkucajte

RECEiVE>receive proba.zip /binary

i datoteka proba zip će biti prebačena iz RECEIVE balera na vaš disk Problem nastaje šlo VAX na kraj svakog bloka dodaje jedan <CR> (carriage return) karakter Da biste ga izbacili morale proći kroz malo komplikovaniju procedure (u primeru je proba zip naka vaša datoteka)

$ ANALYZE /RMS /FDL proba.zip
$ EDiT /FDL proba

i ući ćete u FDL editor gde trabate promeniti CARRIAGE_CONTROL u NONE. tako šlo ćete ići kroz sledeće menije

Modify – Record – CArrigage_controi – None – Ctri-Z – Exit.

Sada vam još preostaje

$ CONVERT /FDL=
proba.fdl proba.zip proba.zip

i biće kreirana datoteka proba zip 2 Obrišala nepotrebne datoteke (proba.fdl 1 proba.fdl.2 i proba zip 1) a proba zip 2 prebacite na vaš PC kao binarnu datoteku

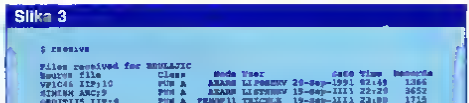
Drugi način je mnogo lakši – končenje posebnog programa TREC Ukoliko koristite UBBG nod DECNet a, ubacite sledeci red u vaš LOGIN COM

$ TREC ==
@ EARN$USERS [PUBLiC]TREC

a za druge nodove se raspitajte kod vašeg sistem administratora. Zatim startujte TREC i pojavice vam se meni Opcijom 1 možete pogledati spisak prisliglih datoteka a sa 3 ih iskopirali iz *receive* balara na disk pri čemu će TREC automatski uraditi konverzije za koje bi vam inače bio potreban goreopisani poslupak

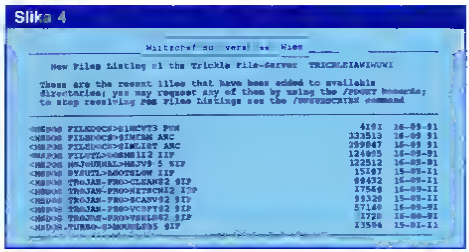
Zanimljivo je da se daloteke koje se nalaze u RECEIVE balaru ne računaju u vašu disk kvotu Ovo je vrlo korisno kad naručujete veću količinu daloteka, a nemate dovoljno slobodnog prostora na disku Kad vam stignu programi mozete ih jedan po jedan prebacivati iz balara na disk (naravno pod uslovom da pojedinačne datoteke nisu duže od slobodnog prostora) pranati na svoj PC pa obnsatl da oslobode prostor za sledeći. Datoteka neće ostati večito u baferu – posle izvesnog (dovoljno dugačkog) vremena biće izbrisane

Slika 3



Primer datoteka pristiglih u RCEiVE bafer

Slika 4



Obaveštenje o novim datotekama

I kad smo već kod disk kvota da pomenemo još jednu prednost končenja programa TREC Pošto se konverzija datoteka (ukidanje <CR> na kraju blokova) vrši automatski prilikom kopiranja iz RECIEVE bafera na disk onda ne morale imati dovoljno praznog prostora za smeštanja dve datoteke koje primate (jedne iskopirane iz balera i druge ispravne dobijene posla CONVERT poslupka)

NEKOLIKO SAVETA

Trebaće vam ipak nesto vremena da se prilagodile *Trickle*-u i naučile da brzo pronađela ono šta vam je potrebno Zato evo još nekoliko saveta za rad i opisa još nekih *Trickle* komandi

Mana **/pddir** naredbe je ta šlo cele samo dobili spisak programa bez ikakvog opisa njihove namene Za lo ce vam biti od konsti datoteke naziva 00-index txt koja se nalaze u većini MSDOS direktorijuma U njima je uz standardne podalke (naziv datoteke dužina i datum) dal i opis programa u jednom redu (obično dovoljno informativan)

Mozete narucrti i spisak (sa opisima) svih daloleka u MSDOS direktorijumima On se nalazi u okviru daloleke (dugačke oko 300 KB) <msdos filedocs>simibm arc Podaci (daloteka simibm idx) su dati u ASCII obliku sa poljima (naziv daloteke duzina datum opis) odvojenim zarazima pa se mogu na primer iz dBASE-a vrlo lako konvertovati u DBF format posle napravite praznu bazu potrebne strukture a zatim zadate

APPEND FROM simibm idx DELiMITED

Ukoliko imate nameru da redovno pratile šta stize u naku od *Trickle* direktorijuma savetujemo se da se pretplatite na listu novopristiglih daloteka Ovo ćete postići ako *Trickle*-u pošaljete komandu

/sub <direktorijum>

npr. **/sub <msdos>** (sub je skraceno od subscribe=pretplatiti) Svakih nekoliko dana dobijacete spisak nelk onome na slici 4 Ukoliko više ne želite da dobijate listu pošaljite komandu **/unsub <direktorijum>**

HERKULES MEKOG SRCA

Uobičajeno je mišljenje da ne postoji način da se na Herculesu softverski definišu „naša" slova. Rešenje ipak postoji, iako ne tako jednostavno kao kod drugih kartica.

Dragan Zakić

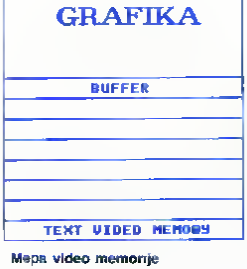
Hercules kartica je stekla veliku popularnost prvenstveno zahvaljujući svojoj ceni. Međutim, mnogima je naneła mnogo muka nemogućnošću da se softverski definiše font koji se prikazuje u tekstualnom režimu. YU slova su većiti problem i na tastaturama i na štampačima i na grafičkim karticama. Razmotrimo mogućnosti da Hercules ipak „progovori ćirilicom".

Naravno, hardversko rešenje mozda i jeste najkvalitetnije rešenje, ali ugraditi naše znakove u EPROM znači zauvek se odreci drugih znakova umesto kojih su naši slovi. Najrasprostranjeniji YUSCII raspored ne odgovara programerima, jer sigurno nije prijatno videti recimo „ašic umesto „ajl, a opet najviše je u upotrebi i zvanično je standardizovan. Softversko rešenje obično je praktičnije, jer vas ničim ne sputava da konsolu raspored koji želite da gepo potrebi isključite ili da pak odlerate znak koji vam se ne svida. Videćemo kasnije zašto da ne, da je isto toliko jednostavno napraviti ćirilični font, koji nije ništa teže aktivirati od „obične" YU latinice.

Herculesu nije moguće saopštiti da želimo da promenimo font i oslaviti ga da sam to obavi kao kod drugih grafičkih karti (VGA). Ako želimo da na ekranu vidimo znak koga nema u ROM-u Herculesa, moramo da aktiviramo grafički prikaz i u video memoriju upišemo poprilično više informacija nego što je potrebno u tekstualnom režimu. Dakle, treba odmah da se pomirimo da to ne može biti tako brzo kao što je u tekstualnom režimu, ali videćemo da se od ove ideje ipak može napraviti nešto korisno.

OSNOVNA IDEJA

Poznato je da Hercules ima 64K video memorije, počev od segmenta B000 od BFFF. Tekst režim koristi 80×25×2=4000 bajtova na samom početku video memorije. Grafički režim ima dve stranice od po 32K, koje počinju na segmentima B000 i B800. I koje možemo po želji aktivirati. Na programiranju Hercules kartice nećemo se zadržavati – o tome možete pročitati u članku „Programiramo Hercules" u „Računarima" 55.



Mapa video memorije

Ideja je sledeća: aktiviraćemo drugu grafičku stranicu, tako da prva ćelija kljobajla na segmentu B000 ostanu slobodna. Pustićemo aplikaciju da u pomenuta ćelija kljobajta bilo DOS i BIOS pozivima bilo direktnim prislupom video memoriji upisuje ono što je potrebno da se prikaže a naš program će „u pozadini" vredno motiti na svaku promenu i sve to prepisivati u grafičku stranicu, ispisujući svakako onlom koji smo mi definisali. Za ovo nam je potreban jedan baler u koji ćemo beležiti šta smo u stvari upisali u grafičku stranicu da ne bismo vise puta radili isti posao. Nema zaista nikakvog razloga da ovaj baler u ciju ušteđe prostora ne smestimo upravo u neiskorišćeni deo video memorije. Kako izraka kaže „kad nešto upišes u memoriju zapamti gde si to slavio" neka naš bafer bude u poslednja ćelija kljobajta prvih 32K video memorije, dakle segment B700.

Program će, dakle, biti rezidentan i da bi se izvršavao treba preuzeti interapt 8 koji generiše tajmer 18.2 puta u sekundi. Posto INT 08h ima najviši prioritet u sistemu, na bi bilo ni malo zgodno onemogućili ostale uređaje da prekidaju naš program kome može zatrebati i malo više vremena da osveži ekran kada dođe do velikih promena (skrol na primer). Zato ćemo sacekali da se izvrši ceo interapt 8, koji će jednim **out 20h,20h** signalizirati interapt kontroleru da je interapt opsluzen i da može generisati sledeci interapt po prioritetu, pa tek onda krenuti u osvežavanje ekrana. Ovo možda izgleda suvišno napominjati ali bisle se pogledom na spisak interapta primetivši da se INT 15h poziva iz osmice i ne radi nista, verovatno zaleteli i prikačili svoju rutinu upravo za njega. Ovakvi pokušaji dovodili su do pojave „gutanja" znakova kod komunikacije sa modemom a verovatno i sa drugim uređajima. Prilikom osvežavanje ekrana nije dozvoljavalo da se znak prislgao na COM port uredno odloži u bafer već su znaci bivali nepovratno izgubljeni.

Treba znati upoređili 80×25=1000 reči sa segmenle B000 (originalna tekst video memorije) i B700 (naš baler) i pri konstalovanju bilo kakve razlike upisati odgovarajući znak u grafičku stranicu i ažurirati bafer. Ovim podrazumevamo da će programi koji se izvršavaju ponašati kao da konsle tekstualni režim i neće pokusavati da aktiviraju grafiku niti pokušavati da iz bilo kog razloga prebace Hercules u tekst režim. Ovo na prvi pogled izgleda vrlo logično i ne zvuči kao neki narocito strog kriterijum. Medutim, prvo startovanje programa zanemarivši ovaj kriterijum, pokazalo je da neke licne programa iz nepoznatih razloga na svom početku (ili kraju ima jedan poziv video BIOS a koji reseluje video karticu što Herkules vraca u tekst režim. Ovakvo ponašanje nekih programa pravi velike probleme i kod definisanja YU slova na VGA karti.

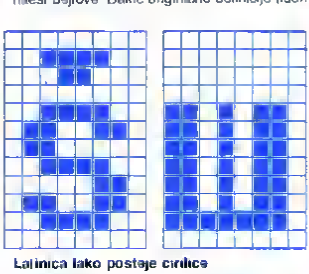
Prvi uslov je razumljiv, jer ne znamo koju će grafičku stranicu program aktivirati, naš program će neumorno prepisivati ažurirati bafer iako se na segmentu B000 i ne nalazi tekst već je taj deo memorije sada organizovan u bit mapiranu sliku. Razni programi se pričino svojevoljno ponašaju pri aktiviranju grafike i na osnovu podataka dostupnih jednom rezidentnom programu nije moguće zaključiti da li je „aktivna grafika ili tekst (kod nas će neravno uvek biti aktivna grafika). Zato ćemo osim osmice preuzeti i interapt 9, koga generiše tastatura kod svakog pritiska na taster i omogućili korisniku da odgovarajućom kombinacijom tastara uključi odnosno isključi naš program.

Drugi uslov je neophodan, jer nam nista na vredi da pišemo po grafičkoj stranici ako je neko reselovao karticu vratio nas u tekst režim i naših znakova ponovo nema. Kako da sprečimo reselovanje Herculesa? Preuzecemo interapt 10h i proveriti kod svakog poziva da li se radi o funkciji 0 (**set video mode**). Ako se radi upravo o postavljanju moda, jednostavno ćemo ignorisati zahlev, obrisati ekran i postaviti kursor u gornji levi ugao.

DEFINISANJE FONTOVA

Ostaje jos da odredimo veličinu fonta. Tekstualni režim koristi font veličine 9×14 piksela koji je pričino nezgodan za prikazivanje u grafičkom režimu iz prostog razloga što je bit mapirana slika organizovana u bajtove (8 bita). Upravo to bi odvlačilo dosta vremena da se šifrovanjem i logičkim operacijama AND i OR karakter ispise na odgovarajuce mesto. Dakle, koristićemo širinu od 8 piksela a za visinu ćemo uzeti 12, jer sa 14 ne bi bio vidljiv ceo poslednji red. Naravno, pri ispisu karaktera vodićemo racuna i o atributu znaka. Inverzni karakter (atributi X111×000) ćemo xorovati vrednošću FFh, ze underline (atributi X000×001) ćemo naknadno dodati crticu na dnu a brighi (atributi XXXX1XXX) ćemo prikazati kao bold tako što ćemo svaki od 12 bajtova koji predstavljaju definiciju znaka zerotirati u desno i operacijom OR sa prvobitnom vrednošću dobiti zadebljani znak. Takode, kada je pozadina crna i boja slova crna (atributi X000<000) znak uopšte ne treba prikazati. Odrećićemo se treparenja karaktera (blink, atributi preko 128) koje se jonako vrlo retko koristi u programima za obradu teksta, gde ćemo program najviše koristiti.

Kako je organizovana definicija fonta u datoteci? Za jedan karakter potrebno nam je dva naesi bajtove. Dakle, originalne definicije (iden-



Latinica lako postaje ćirilica

Listing 1

```
newkeyint:
        pusha
        call dword ptr [oldtimer]       ;poziv original interrupta
        sti                             ;dozvoli prekide
        cmp ss:[ucuvaj_se],1            ;da li je pustena skripta
        jne ireturn                     ;ako nije  izlaz
        cmp cs:[brojac],1               ;da li je vreme za pocetak rolovanja
        jne cs:[brojac],1               ;ako jeste  signalizacija da je vreme za rolovanje
ireturn:
        iret                            ;povratak iz interrupta

begin:
        mov ax,0040h                    ;adresiranje u bios
        mov es,ax                       ;segment 40h
        push es:[6ch]                   ;stara vrednost brojaca
        pop word ptr cs:[brojac]        ;postavljanje brojaca
        mov ax,cs:[6ch]                 ;stari interrupt
        mov word ptr [oldtimer],ax      ;postavljanje novog interrupta
        mov ax,cs
        mov es:[6ch],ax
        mov ax,0
        mov ds,ax                       ;segment 0
        mov ax,[4*10h]                  ;stara adresa int 10h
        mov word ptr [oldint],ax        ;postavljanje novog int 10h
        mov ax,cs
        mov [4*10h],ax
        mov ax,offset newint
        mov [4*10h+2],ax
        mov ax,0b800h                   ;segment video memorije
        mov es,ax
        mov di,0                        ;offset 0
        mov cx,2000                     ;2000 karaktera
        mov al,' '                      ;karakter space
        mov ah,07h                      ;atribut
        cld                             ;smer ka visim adresama
        rep stosw                       ;ispis space-ova
        mov ah,02h                      ;funkcija za pozicioniranje kursora
        mov bh,0                        ;video page 0
        mov dx,0                        ;pozicija 0,0
        int 10h                         ;poziv interrupta
        mov ax,0b800h                   ;segment video memorije
        mov es,ax
        mov di,0                        ;offset 0
        mov cx,2000                     ;2000 karaktera
        mov al,' '                      ;karakter space
        mov ah,07h                      ;atribut
        cld                             ;smer ka visim adresama
        rep stosw                       ;ispis space-ova
        mov ah,02h                      ;funkcija za pozicioniranje kursora
        mov bh,0                        ;video page 0
        mov dx,0                        ;pozicija 0,0
        int 10h                         ;poziv interrupta
        mov ax,3100h+((offset end_of_program-offset begin+15)/16)
        int 21h                         ;terminate and stay resident

oldtimer        dd      ?               ;stara adresa int 1Ch
oldint          dd      ?               ;stara adresa int 10h
brojac          db      ?               ;brojac pauze
ucuvaj_se       db      0               ;da li je pustena skripta
end_of_program:
```

Listing 2

```
check_installed:
        mov ah,installed_id             ;provera da li je YUHERC vec instaliran
        mov es,0
        mov bx,0150h                    ;adresa int 54h
check:
        inc bx                          ;sledeca adresa
        cmp es:bx,'YU'                  ;provera da li je
        jne not_installed               ;ako nije  nije instaliran
        inc bx                          ;sledeca adresa
        cmp es:bx,'HE'                  ;provera da li je
        jne not_installed               ;ako nije  nije instaliran
        inc bx                          ;sledeca adresa
        cmp es:bx,'RC'                  ;provera da li je
        jne not_installed               ;ako nije  nije instaliran
        mov ah,0ffh                     ;ako jeste  instaliran
not_installed:
        or ah,ah                        ;provera da li je instaliran
        jz installed                    ;ako jeste  skok na kraj
        mov ah,09h                      ;ako nije  ispis poruke
        mov dx,offset copyright_msg     ;offset poruke
        int 21h                         ;poziv interrupta
installed:
        mov al,ah                       ;povratna vrednost
        mov ah,4ch                      ;funkcija za terminate
        int 21h                         ;poziv interrupta
copyright_msg   db      'Copyright (C) 1991 YUHERC$'
```

Rutine za prepisivanje znaka u gralicku stranicu

Rutina za proveru da li je YUHERC vec instaliran

ličnе onima u ROMu) svih 256 karaktera smeštićemo sukcesivno od nultog do 255-og. Za njih ćemo staviti definicije naših karaktera i to sledećim redoslednom: ČćđšŽČĐŠ. Konstimo ovaj način jer nam je sada dostupno svih 256 znakova + 10 naših i nakon startovanja programa ih lako možemo iskoristiti na željeno mesto u zavisnosti od kodnog rasporeda koji se koristi. Na ovakav ce način biti smešten i ioni u samom programu da bi ga jednostavno zamenio drugi sa diska ako je potrebno. Kada korisnik preko komandne linije zatraži određeni kodni raspored definiciju naših slova postavićemo na odgovarajuće mesto u zavisnosti od zahtevanog kodnog rasporeda a stare definicije slavićemo na mesto naših slova. Izvršićemo dakle zamenu da bismo kada buda zatražen drugi kodni raspored mogli da definiciju vratimo na svoje mesto. Ovaj posao obavlja program **exchange**.

Treba još da rešimo problem prikazivanja kursora. Pošto cela stvar radi u grafičkom režimu, kursor možemo prikazati kako nam je volja. Mnogi nalaze da je prijatniji slalični kursor u obliku inverznog karaktera od uobicajenog. Prepuceg pa cemo se opredelili za Ju varijan Ju. Naravno za one koji više vole trepcuci kursor ostavićemo da se opcijom može aktivirati. Za treperenje konstimo jednu memorijsku promenljivu koju interapt 8 svaki put uveca za jedan. Maskiranjem jednog njenog bita dobijamo uslaljeno treperenje odnosno taj bit se naizmenično pali i gasi. Od maskiranog bita zavisi i brzina treperenja kursora. Ie lako opcija **-f1** aktivira najbrže treperenje a **-f4** najsporije. Opcija **-f0** isključuje treperenje kursora.

ASEMBLER JE BOLJI

Pošto program treba da bude rezidentan dakle biće stalno prisutan i zauzimali svojih **x** kilobajta memorije. Treba se potruditi da lo **x** bude što je moguće manje. Poznato je da viši programski jezici nisu naročito štedljivi u pogledu zauzeca memorije. Drugi razlog da se opredalimo za asembler je brzina koju rutina ova kve namene mora da ima. I zaista teško da bi se u C-u mogao napisati sličan program koji bi se po veličini makar približio ovom koji je za jedno sa definicijom fontova dugacak nekih 5K. Naravno nece sve od loga ostali rezidentno. Deo programa neophodan je za analizu komandne linije postavljanje parametara i još nake provere. Taj deo nam kasnije neće trebati i nema potrebe da bez razloga zauzima memoriju.

Na početak našeg programa stavićemo proveru da li je vec instaliran da nekim slučajem ne bismo u memoriji nepotrebno napravili dve kopije istog programa. S druge strane moze biti korisno naknadno kada je program vec instaliran menjali parametre koji su postavljeni pri instalaciji. Na primer moze se birati između 5 kodnih rasporeda koji su dostupni na Sezamu. Naravno dodavanje novih kodnih rasporeda nece biti teško na osnovu priloženog listinga. Pored promene kodnog rasporeda omogućićemo da se umesto poslojeceg ucita novi

Nastavlja se na strani 51

POPUNJAVANJE FORMULARA

U prethodnim prilozima obuhvatili smo dva vida prikaza podataka na ekran: funkcije visokog nivoa (dijalozi kroz potprozore) i funkcije nižeg nivoa, sa neposrednim grafičkim izlazom. Novi prilog o programiranju prozora ukazuje na još jedan vid formiranja grafičkog izlaza u *Windows* programima – izlaz na štampač. I ovaj prilog potkrepićemo programskim primerom.

Dejan Dinić

Wform je program za popunjavanje formulara – obrazaca. Iz unapred pripremljene datoteke program štampa podatke na zadate pozicije na odštacu. Sa korisničke strane „krasi" ga nekoliko tipičnih osobina *Windows* programa. Opis polja na obrascu je interaktivan učitavanjem u koordinatnu mrežu na ekranu sve u dobrom WYSIWIG maniru. Vizuelno zadavanje polja u razmeri 1:1 treba da nas oslobodi mnogobrojnih proba sa štampačem, ušteđi vreme i smanji gomilu probnih primeraka formulera.

OPIS OBRASCA

Obrasci koji su u upotrebi uglavnom ne vode računa o karakteristikama štampača, pa je često prilično neslandaran. Program zato omogućuje postavljanje polja u grafičkim koordinatama.

Podaci se učitavaju iz obične tekst datoteke. Polja su razdvojene posebnim znakom (delimiter). Pretpostavljeni znak je „." Nanizena polja jednog obrasca treba završiti oznakom kraja reda (CR/LF).

Opis programa počećemo od načina na koji se formira opis obrasca. Obrazac je u programu predstavljen kroz strukturu *Form* (listing 1). Ona sadrži promenljive koje opisuju širinu i visinu obrasca, broj polja, i niz sa osobinama svih polja. Struktura se može snimiti u datoteku i kasnije učitati. Modifikacija obrasca se vrši u obradi poruka o kretanju miša (WM_MOUSEMOVE) i pritisku (WM_LBUTTONDOWN) i otpuštanju (WM_LBUTTONUP) tastera na mišu (listing 2). Iako je za *Windows* tipično samo korišćenje levog tastera na mišu (i naš primer se drži toga), napominalmo da je programski moguće potpuno jednako tretirati i srednji (ako ga ima) i desni taster. Funkcije **StartMark** **DoMark** i **EndMark** obrađuju ove poruke i beleže opise polja u nizu **Form.Fids**.

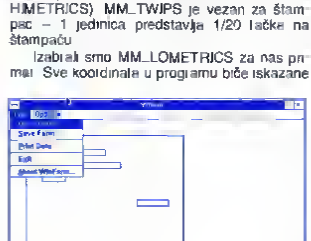
JEDINICE MERE

Rekli smo već da je za ovaj primer bitan odnos između obrasca i njegove grafičke predstave na ekranu. *Windows* funkcije koje će nam pomoći u tome pripadaju grupi funkcija za mapiranje. U prethodnim prilozima pišući po ekranu (bolje reći, crtajući tekst) nismo se previše osvrtali na to kakve će koordinate konstruisati kakav je to uopšte koordinatni sistem. Moglo se uočiti da je koordinatni početak bio u gornjem levom uglu radne površine prozora i da je **x** koordinata rasla na desno, a **y** na *dole*. Jedinica mere u tom sistemu bila je ekranska tačka (*pixel*). To je samo jedan (MM_TEXT) od mogućih koordinatnih sistema – načina mapiranja. Ostali se mogu podeliti u dve grupe.

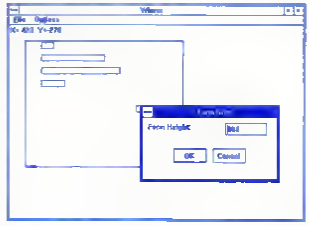
U prvoj grupi su MM_ISOTROPIC i MM_ANISOTROPIC. Glavna njihova osobina je da programeru dozvoljavaju da odredi orijentaciju osa i razmeru. Razmera u ovom kontekstu predstavlje odnos između logičkih koordinata i fizičkih jedinica (tačke ekrana, na primer). Najma sve izlazne rutine koriste logičke koordinate. Pre ispisa na uređaj one se shodno postavljenom načinu mapiranje preračunavaju u fizičke koordinate konkretnog uređaja. Uz korišćenje gore pomenutih načina mapiranja, može se u programu uspostaviti logički crtaći prostor dimenzija 20000 * 10000 (logičkih jedinica), a da jo na ekranu predstavlja tek četvrtinu radne površina prozora. U takvom sistemu linija od nekoliko stotina jedinica bila bi na ekranu dugačka svega pet tačaka. Prostom izmenom parametara može se povećati ili smanjiti realna površina na ekranu koja predstavlja logički prostor, a time i stvarna veličina crteža u tom prostoru. U logičkom prostoru crtež ostaje isti i program „misli" da crta istu stvar.

Druga grupa sadrži nekoliko načina mapiranja sa fiksnom orijentacijom i razmerom. Tu su MM_LOMETRICS MM_HIMETRICS, MM_LOENGLISH, MM_HIENGLISH i MM_TWIPS. Orijentacija je kod svih ista **x** raste na desno, a **y** na gore. Koordinatni početak je u gornjem levom uglu prozora što znači da je vidljivi deo sistema negativna **y** koordinata. Razmere se mogu naslutiti iz naziva sufiks ENGLISH govori da su logičke jedinice delovi inca (0.01 za LOENGLISH, 0.001 za HIENGLISH), a METRICS u nazivu nas upućuje na metar (0.1mm za LOMETRICS i 0.01mm za HIMETRICS). MM_TWIPS je vezan za štampač – 1 jedinica predstavlja 1/20 tačke na štampaču.

Izabrali smo MM_LOMETRICS za nas primer. Sve koordinate u programu biće iskazane u desetinama milimetra. U praksi se primećuju izvesna (sasvim razumljiva) odstupanje na nekoliko VGA monitora uočili smo do 10% grešake što manje.



Osnovni meni aplikacije



Dijafog za unos veličine obrasca

Jako izlazne funkcije koriste logičke koordinate, kod miša nailazimo na malo odstupanje od ovog sistema. Poziciju miša sistem daje u fizičkim koordinatama. Za konverziju se koristi funkcija **DPtoLP** (DevicePointtoLogicalPoint). Postoji i funkcija sa inverznim procesom – **LPtoDP**.

ŠTAMPANJE OBRASCA

Pošlo smo definisati obrazac, ostaje nam da se posvetimo štampi. Kao što ekran ima svoj *display context* – kontrolnu strukturu sa opisom prozora u koji se crta. Jako je i za štampač potrebno formirati ovakvu strukturu (listing 3). Hendl koji je identifikuje ne razlikuje se on onog za ekran. Tako se za crtanje i pisanje na štampaču koriste iste funkcije kao i za ekran, uz hendl kao argument. Da stvari ipak ne bi bile jako jednostavne, za razliku od ekranskog DC-a koji je ju po prirodi stvari (prozor se negde mora prikazati) i povezan vilo čvrsto sa osnovnim komponentama *Windowsa* veze sa štampačem je malo slobodnije. Štampač ne morate uopšte imati ili ih možete imati više. Zato je za formiranje DC-a štampača potrebno izvestan uvod. Naša funkcija **GetPrinterDC** preuzima parametre sistema koji su zadati u konfiguracionoj datoteci *Windowsa* – WIN.INI – i odatle uzima osnovne podatke o štampaču. Potrebno je još postaviti proceduru koja omogućava prekid štampe – kod nas je to novi prozor (child window) sa prepoznatljivim „Cancel" tasterom.

Uvodni deo štampe još nije gotov – potrebno je umetnuti *I spooler*. U saradnji *Windows* sistema ulazi i *PrintManager*. Sve što se šalje na štampač, prvo se zapravo slavi u red čekanja koji kontroliše *PrintManager* i odatle (u pozadini) se šalje na sam uređaj. Kako je *Windows* višeprogramski sistem i kako je moguće da više programa u isto vreme šalje nešto na štampač potrebno je uvesti malo reda. Početak štampe programa označava slanjem koda STARTDOC sa nazivom posla (*job*) koji se štampa. Tek nakon toga slanje stvarnog sadržaja ispisa može da počne. Kada je završen ispis potrebno je još oznaciti kraj posla porukom ENDDOC. Konačno to je dovoljno da *PrintManager* zaključi predmet i ako je štampač slobodan, počne sa „stvarnom" štampom.

Kontrolni kodovi i sve instrukcije koji na spadaju u izlazne grafičke funkcije a služe kontroli štampača prosleđuju se konačnom funkcije **Escape**.

Deo programa koji učitava podatke iz tekst datoteke je napisan standardno. Interesantno je da je do verzije 3.0 u *Windows* programima bilo zabranjeno korišćenje **C-run time stream** I/O funkcija (verovatno zbog dinamičkih alokacija memorije na ne-*Windows* način). U dokumentaciji za verziju 3.0 sa kaže da se tako sa

Listing 1

```
struct
{
    int  X_EXT;          /* sirina obrasca   */
    int  Y_EXT;          /* visina obrasca   */
    int  nFlds;          /* broj polja       */
    struct
    {
        int  x;
        int  y;
        int  xs;
        int  ls;
    } Flds[MAX_FLDS];    /* definicija polja */
} Form;
```

Struktura koja opisuje obrazac

Listing 2

```
/*--------------------------------------------------------------------*/
/*  obrada poruka koje stizu glavnom prozoru aplikacije               */
/*--------------------------------------------------------------------*/
long FAR PASCAL MainWndProc(hWnd  message  wParam  lParam)
HWND hWnd;
unsigned message;
WORD wParam;
LONG lParam;
{
    HDC         hDC;
    PAINTSTRUCT ps;
    RECT        Rect;
    HANDLE      hMenu;
    FARPROC     lpProcAbout;
    PAINTSTRUCT ps;

    switch (message) {
        case WM_COMMAND:
            switch (wParam) {
                case IDM_ABOUT:
                    lpProcAbout = MakeProcInstance(About, hInst);
                    DialogBox(hInst, "AboutBox", hWnd, lpProcAbout);
                    FreeProcInstance(lpProcAbout);
                    break;

                case IDM_OPEN:
                    OpenForm(hWnd);
                    break;

                case IDM_SAVE:
                    SaveForm(hWnd);
                    break;

                case IDM_SETUP_WIDTH:
                    SetFormWidth(hWnd);
                    break;

                case IDM_SETUP_HEIGHT:
                    SetFormHeight(hWnd);
                    break;

                case IDM_PRINT:
                    PrintForm(hWnd);
                    break;

                case IDM_EXIT:
                    DestroyWindow(hWnd);
                    break;

                }
                break;

            case WM_PAINT:
                BeginPaint(hWnd, (LPPAINTSTRUCT)&ps);
                MinimizePaint(hWnd, ps.hDC);
                EndPaint(hWnd  (LPPAINTSTRUCT)&ps);
                break;

            case WM_SIZE:
                InvalidateRect(hWnd  NULL  TRUE);
                break;

            case WM_LBUTTONDOWN:
                if (mode == WORD_IDLE)
                {
                    hDC = GetDC(hWnd);
                    StartMark(hWnd, hDC, lParam);
                    ReleaseDC(hWnd, hDC);
                }
                break;

            case WM_MOUSEMOVE:
                hDC = GetDC(hWnd);
                if (mode == WORD_MARK)
                    DrawMark(hWnd, hDC, lParam);
                SetMouseCursor(hDC);
                ReleaseDC(hWnd  hDC);
                break;
            case WM_LBUTTONUP:
                if (WORD == WORD_MARK)
                {
                    hDC = GetDC(hWnd);
                    EndMark(hWnd, hDC, lParam);
                    ReleaseDC(hWnd, hDC);
                }
                break;

            case WM_DESTROY:
                PostQuitMessage(0);
                break;

            default:
                return (DefWindowProc(hWnd, message, wParam, lParam));
        }
        return (NULL);
    }
```

Funkcija za obradu poruka glavnog programa

Listing 3

```
/*--------------------------------------------------------------------*/
/*  Inicijalizacija i petlja za stampu obrasca                        */
/*--------------------------------------------------------------------*/
int PrintForm (hWND)
HWND hWnd;
{
    HDC   hPr;
    int   i, IoStatus;
    char *txtfld;
    BOOL *f;

    hSaveCursor = SetCursor(hHourGlass);
    hPr = GetPrinterDC();
    if (!hPr)
        MessageBox(hWND, "Greska u stampi"  NULL, MB_OK | MB_ICONHAND);
        return (0);
    }

    lpAbortDlg =  MakeProcInstance(AbortDlg, hInst);
    lpAbortProc = MakeProcInstance(AbortProc, hInst);

    /* onemogući fokusiju za prekid */
    Escape(hPr   SETABORTPROC  NULL,
           (LPSTR) lpAbortProc,
           (LPSTR) NULL);

    /* Javi spooleru za početak štampe */
    if (!Escape(hPr, STARTDOC  10  "Form1 test",
                (LPSTR) NULL) < 0) {
        MessageBox("Greska start print"
            NULL  MB_OK | MB_ICONHAND);
        FreeProcInstance(lpAbortDlg);
        FreeProcInstance(lpAbortProc);
        DeleteDC(hPr);
    }

    bAbort = FALSE;      /* indikator prekida */

    /* Kreiraj Abort dialog box */

    hAbortDlgWnd = CreateDialog(hInst, "AbortDlg",
                   hWND  lpAbortDlg);

    if (!hAbortDlgWnd)
        SetCursor(hSaveCursor);
        MessageBox(hWnd  "Nemoguce kreiranje dijaloga",
            NULL, MB_OK | MB_ICONHAND);
        return (FALSE);
    }

    /* Prikazi hBox dialog                    */
    ShowWindow(hAbortDlgWnd, SW_NORMAL);

    /* Iskljuci prijem poruka u main window */
    EnableWindow(hWnd, FALSE);
    SetCursor(hSaveCursor);

    /*  pronadji stampacu dimenzije strane   */
    GetFormSize(hPr);

    TextOut (hPr, 0, 0, (LPSTR) "Start", strlen("Start"));
    SetMapMode (hPr, MM_LOMETRIC);

    /*  Zapis polja u obrazac   */

    f = fopen (DataFile, "r");
    while (fgets(sz, MAX_LINE, f) != NULL)
    {
        txtfld = strtok(sz, delimiter);    /*  izvadi polje  */
        for (i=0; i<Form.nFlds && txtfld != NULL; i++)
        {
            TextOut(hPr,
                Form.Flds[i].x,
                Form.Flds[i].y,
                (LPSTR) txtfld,
                strlen(txtfld));
            txtfld = strtok(NULL, delimiter);
        }
        /* Escape(hPr, NEWFRAME  0, 0L, 0L); */
    }
    fclose (f);

    /* zakljuci stampu, ako nije vec doslo do prekida stampe */

    if (IoStatus >= 0 && !bAbort) {
        Escape(hPr, NEWFRAME  0, 0L, 0L);
        Escape(hPr, ENDDOC  0, 0L, 0L);
    }
    EnableWindow(hWnd, TRUE);

    /*  poništi sa sobom  */

    DestroyWindow(hAbortDlgWnd);
    FreeProcInstance(lpAbortDlg);
    FreeProcInstance(lpAbortProc);
    DeleteDC(hPr);
}
```

Funkcija za štampanje

sada te funkcije mogu konstituisati prekidanje preporucuje upotreba funkcija nižeg nivoa (**read write**) U ovom slučaju odučili smo se **stream** funkcije na sopstvenu odgovornost Klasičnom petljom čita se po jedan red datoteke, iz njega izdvajaju polja i štampaju

otvorena samo dok nad procesorom – za vreme obrade jedne poruke Ako to nije moguce uskladiti sa logikom programa u slucaju kakav je na primer u *Worm* programu kada se kontrola gubi otvaranjem nekog potprozora, preporučuje se da

Treba skrenuti pažnju na rad sa datotekama u višeprogramskom okruzenju Jedna od mnogobrojnih preporuka u dokumentaciji nas upucuje na to da ako je ikako moguce datoteke treba drzati

taj potprozor ima ekskluzivnu kontrolu fokusa (**system-modal dialog**) Ovde smo se našli u malo dilemi ako potprozor za (eventualno) prekidanje štampe proglasimo ekskluzivnim naš program na može raditi u pozadini (*background*) što baš nije pohvalno za jedan program za štampu (i to u višeprogramskom okruzenju) Odslupili smo od preporuke pretpostavljajući da korisnik ovog programa neće modifikovati podatke u toku štampe Program tako ostaje upotrebljiv za rad u pozadini Nadamo se da ce vam dilema koju smo imali poslužiti kao korisna ilustracija koncepcijskih problema na koje se nailazi pri prelasku na visepiogramski sistem rada

APSTRAKTNE KLASE

U ovoj seriji tekstova bavili smo se glavnim osobinama jezika C++. Veliki deo jezika C++ poklapa se sa jezikom C i te, zajedničke, osobine ovde nećemo posebno objašnjavati. C++ uvodi niz novina u odnosu na sam C i uzdiže ga na nivo objektnog jezika. Skoro svaki C program će se prevesti i pod bilo kojim C++ sistemom, ali obratno ne važi. Poslednji nastavak serije posvećen je najtežim stvarima – apstraktnim klasama i prenosu parametara.

Duško Savić

ODLOŽENO POVEZIVANJE

Datoteka A49.CPP sadrži test za tri virtuelne klase (program je dat u prošlom broju) pridodeljuje vrednosti i štampa ih

Listing a49

```
#include <stdio.h>
#include "a40.cpp"
sfera loptica("loptica", 1.0);
kocka kristal("ugljenik", 1e-24);
opis * oblici[] = {
        &loptica,
        &kristal,
};
main(){
        printf("\nProgram A49.CPP \n");
        // stampa pojedinacno opise
        loptica.print();
        kristal.print();
        // stampa sve opise iz liste
        for(int i = 0; i < sizeof(oblici) / sizeof(oblici[0]); i++)
          oblici[i]->print();
}
```

Test za virtuelne funkcije

Rezultati su

```
Program A49 CPP
loptica poluprecnik = 1
ugljenik
ivica duzina = 1e-24
loptica
ugljenik
```

Poziv pridružene funkcije **print()** kao poruke konkretnom objektu štampa kompletnu informaciju o objektu. Ij. izvršavaju se metodi iz klasa **sfera** i **kristal**. Međutim, kada se objekat definiše kao primerak osnovne klase (primenom funkcije **print()** na sve pokazivače u nizu **oblici**) – izvršava se metod iz osnovne klase **opis**. To uopšte nismo želeli! Idealno bi bilo imati i zajednički interfejs iz osnovne klase i različite efekte unutar svake izvedene klase.

Problem nastaje usled pretanog povezivanja funkcije **print()**: jer se već tokom prevođenja određuje njena izvršna adresa u prevedenom programu. To je i logično, jer jedina funkcija **print()** za koju prevodilac zna je upravo **opis::print()**. Takvo povezivanje se naziva *ranim* ili *statičkim povezivanjem*. Rešenje problema je da nekako naznačimo prevodiocu da ne uzima **opis::print()** već da uzme **sfera::print()** ili **kristal::print()** zavisno od toga da li je objekat kojem se upućena poruka **print()** lipa **sfera** ili **kristal**. To je moguće samo ako lakav objekat postoji – pa pošto objekti postoje tek u vreme izvršenja programa, mora se uvesti *kasno* ili *dinamičko povezivanje*. Na nivou sintakse potrebna nam je nova službena reč – **virtual** (Na engleskom virtual znači „onaj koji je odložen"). Ij. virtuelne funkcije su funkcije sa odlozenim povezivanjem.) Njome označavamo jednu ili više funkcija iz klase. Virtuelnoj funkciji prevodilac „polajno" dodaje pri nek pripadnosti klasi – i to u obliku pokazivača na tabelu pokazivača na funkcije. Tabela virtuelnih poziva sadrži pokazivače na sve funkcije deklarisane kao **virtual** u jednoj klasi odnosno u bilo kojoj izvedenoj klasi. Program a50.cpp pokazuje kako možemo detektovati prisustvo pokazivača na tabelu virtuelnih poziva pomoću **sizeof()**

Rezultat programa A50.CPP je

```
Program A50 CPP
sizeof(nijeVirtualna) = 2
sizeof(jesteVirtualna)  4
```

Prilikom svakog poziva virtuelne funkcije prevodilac generiše mašinske naredbe koje uz pomoć pokazivača na tabelu virtuelnih poziva „proteže" kroz tu tabelu i pozivaju odgovarajuću funkciju. Evo kako treba ispraviti klasu **opis** (A47.CPP) pa da funkcija **print()** bude virtuelna

Bilo je dovoljno dodati samo reč **virtual** ispred **void print()**. Rezultat izmenjenog programa je

Listing a50

```
//      virtuelnih poziva pomocu sizeof()
#include <stdio.h>
class nijeVirtualna{
        int x;
public:
        void bezVeze() {}
};
class jesteVirtualna{
        int x;
public:
        virtual void bezVeze() {}
};
main() {
  printf("\nProgram A50 CPP\n");
  printf("sizeof(nijeVirtualna) = %d\n", sizeof(nijeVirtualna) );
  printf("sizeof(jesteVirtualna( = %d\n", sizeof(jesteVirtualna) );
}
```

Otkrivanje pokazivača na tabelu virtualnih poziva pomocu sizeof()

Listing a47

```
//      klase koje ispisuje objasnjenje na ekran
#include <stdio.h>
#ifndef OPIS1_H
#define OPIS1_H
class opis {
protected: // izvedene klase imaju pristup
  char * informacija;
public:
  opis(char * info) : informacija(info) {}
  virtual void print() { printf("%s\n", informacija); }
};
#endif // OPIS1_H
```

Prvi deo primera za virtualne klase – klasa koja ispisuje objašnjenje na ekran

```
Program A49 CPP
loptica poluprecnik = 1
ugljenik ivica duzina = 1e-24
loptica poluprecnik = 1
ugljenik ivica duzina = 1e-24
```

Bez virtualnih funkcija nije moguće stvoriti hijerarhiju klasa u kojoj se svima obraćamo preko istih opština pravilima a da svako u njoj reaguje shodno svojoj prirodi. Virtuelne funkcije su vrlo česte pa se može postaviti i pitanje zašto sve funkcije nisu virtualne? Mogle bi biti (u smislu sasvim sigurno jesu) ali ovako programer može da bira između ugodnosti u programiranju (što mu pružaju virtualne funkcije) i brzine izvršavanja prevedenog programa (jer se kasno povezivanje upravo ilo odigrava i malo – za jednu kratku poseli tabeli virtuelnih poziva – uspostava programa). C++ programer zna da su mu obe vrste povezivanja na raspolaganju, bira ih sam, već prema situaciji i na svoju odgovornost.

Virtualna funkcija mora biti deklarisana u osnovnoj klasi. Čak i ako je nikad eksplicitno ne zovemo. Ako funkciju nedefinišemo u izvedenoj klasi moramo je i deklarisati iznova. Prototipovi virtuelnih funkcija moraju biti identični niz hijerarhiju klasa. Inače će C++ prevodilac u verzijama 1.2 posleli poruku o grešci. U verziji 2.0 greška neće biti prijavljena, ali će prevodilac podrazumevati da funkcija više nije virtualna. Naravno, prototip nevirtualne pridružene funkcije u izvedenoj klasi možemo menjati po volji. Šta je ekvivalentno deklarisanju nove funkcije.

Ne postoje virtuelni konstruktori ali mogu postojati virtualni destruktori. Takođe je moguće preopterećenje virtualnih operatora

APSTRAKTNE KLASE

Nasleđivanje otvara mogućnosti da se prave apstraktne klase. Ij. klase čije objekte nikada nećemo aktivirati u programu. Apstraktna klasa postoji samo da bi bila nasleđivana. Njen zadatak je da definiše zajednički interfejs i da postavi obrazac za izvedene klase. U pojedinim objektnim jezicima (smoltok, aklol) sve

klase izviru iz jedne zajedničke klase koja se obično zove **Object**. U C++-u programer može svoje naredbe organizovati tako da pripadaju istoj hijerarhiji ali i ne mora. Međutim, postojanje ISIS osnovne klase pruža velike mogućnosti za saradnju velikog broja klasa koje – sa svoje strane – može pisati veliki broj programera i to istovremeno. Kao primer za apstraktnu klasu navodimo konzistentno prijavljivanje grešaka u klasama iste hijerarhije. Klasa **testiranje** je data u datoteci A51.CPP

Listing a51

```
#ifndef TEST1_H
#define TEST1_H
#include <stdio.h>
class testiranje {
public:
  virtual void prikaz() {
    puts("testiranje -- greska: funkcija prikaz nije definisana");
  }
};
#endif TEST1_H

// A52 -  test za klasu testiranje
#include "a51.cpp"
class X : public testiranje{
    int i;
public:
        X(int ii = 0) {
                i = ii;
        }
        void prikaz() {
                printf("i = %d\n", i);
        }
};
class Y : public testiranje {
        float f;
public:
        Y(float ff = 0.0) {
                f = ff;
        }
        /*void prikaz() {
                printf("f = %f\n", f);
        } */
};
main(){
        printf("\nProgram A52 CPP\n");
        X x(1);
        Y y(4.4);
        x.prikaz();
        y.prikaz();
        // x i y su takodje i clanovi klase testiranje
        testiranje * test[2];
        test[0] = &x;
        test[1] = &y;
        test[0] -> prikaz();
        test[1] -> prikaz();
}
```

Apstraktna klasa za testiranje

Rezultat je

```
Program A51 CPP
i = 1
testiranje -- greska: funkcija prikaz nije definisana
i = 1
testiranje -- greska: funkcije prikaz nije definisana
```

Vidimo da je u jednoj klasi funkcija **prikaz()** izmenjena, a da u drugoj nije. Jasno je da se na ISIS način može promeniti kompletan korisnički interfejs programa i to prostom redefinicijom osnovne klase. Izmena u apstraktnoj klasi prenosi se kroz sve naslednike.

C++ u verziji 2.0 ima i posebnu definiciju apstraktne klase. To je svaka klasa koja sadrži bar jednu čistu virtualnu funkciju, gde je „čista" virtualna funkcija ona virtualna funkcija koja se završava sa = 0. Na primer

```
class oblik {
  point centar;

public
  gde() { return centar; }
  virtual void rotacija(int) = 0;      // cista virtualna funkcija
  virtual void crtanje() = 0;          // takode
  virtual void osvetli() = 0;          // takoda
};
```

Naredba popul **oblik x;** bi proizvela grešku.

VIŠESTRUKO NASLEĐIVANJE

Jedna od osobenosti C++-a u odnosu na ostale objektne jezike je višestruko nasleđivanje: klasa nasleđuje dve ili i još više klasa istovremeno. Primer je dat na listingu a53

```
Program A53 CPP
konstruktor za c1
konstruktor za c2
konstruktor za c1c2
ic1 = 3, ic2 = 567
```

Klasa **c1c2** nasleđuje istovremeno i klasu **c1** i klasu **c2**. Što se vidi iz reda

```
class c1c2 : c1, c2 {
```

Listing a53

```
#include <stdio.h>
class c1 {
protected:
        int ic1;
public:
        c1() {
                printf("\nkonstruktor za c1");
        }
        c1(int i1) {
                c1();
                ic1 = i1;
        }
};
class c2 {
protected:
        int ic2;
public:
        c2() {
                printf("\nkonstruktor za c2");
        }
        c2(int i1) {
                c2();
                ic2 = i1;
        }
};
class c1c2 : c1, c2 {
        int ic12;
public:
        c1c2 () : c1(3), c2(567) {
                printf("\nkonstruktor za c1c2");
                printf("\nic1 = %d, ic2 = %d", ic1, ic2);
        }
};
main() {
        printf("\n\nProgram A53 CPP");
        c1c2 C;
}
```

Primer za višestruko nasleđivanje

Konstruktori klase **c1c2** može zadati koji se konstruktori osnovnih funkcija pozivaju. Navodimo ih iza dvotačke koja sledi iza deklaracije konstruktora. Što se vidi iz reda

```
c1c2 () : c1(3), c2(567) {
```

Izvedena klasa ima pristup **protected** internim promenljivama iz svake po jednoj.

Višestruko nasleđivanje je mač sa dve oštrice. Sa jedne strane lepo je spajati klase u stilu „neka ovaj objekat bude i lipa prozora i tipa konkurentnog procesa" ali se time hijerarhijsko uređenje klasa pretvara u mrežu. Mnogim programerima taj prelaz se čini neopravdanim: mreža nije instrument klasifikacije a hijerarhija jeste.

KLASA TEKST

Evo kako bismo mogli da stvaramo nove hijerarhije klasa. Podimo od uobičajenog pojma laksika (*string*). C++ ih podržava na nivou poziva iz zaglavlja **string.h**. Problem sa leksikalnim varijablama je što se za njih memorija zauzima na steku a to su moguće razne „zloupotrebe" npr. neovlašćen upis novih podataka preko jednom definisane laksikalne konstante. Zato za leksikal zauzimamo prostor u dinamičkoj memoriji korišćenjem operatora **new** i kopiramo sadržaj laksika pomoću **strcpy**. Funkcija **strcpy** je ovde zgodna jer premešta i NULL terminator teksta

Listing a54

```
#include <iostream.h>
#include <string.h>
#include <alloc.h>
class Tekst {
protected:
        char * tkst;
public:
        Tekst(char * tekst = " "){
          // kopira tekst kao novi podatak
          tkst = new char[strlen(tekst) + 1];
          strcpy(tkst, tekst);
        }
        operator void*() { return (void *) this; }
        char * cp() { return tkst; }
        friend ostream& operator<<(ostream&s, Tekst & sp) {
                s << sp.tkst;
                return s;
        }
        Tekst() { free(tkst); }
};
```

Klasa Tekst (Tekst)

Naredba

```
tkst = new char[strlen(tekst) + 1]
```

odvaja tačan broj bajtova u dinamičkoj memoriji i određuje se kao dužina laksta uvećana za 1 tj. za NULL terminator. Pokazivač **tkst** pokazuje na početak leksta pa je u destruktoru argument za **free()**

Klasu **Tekst** ćemo nasleđivati pa je izdvojena u posebnu datoteku A54.CPP. Glavni program A55.CPP pokazuje da se laksi definisan na steku može neovlašćeno (ili nenamerno svejedno) pokvariti. Tekstu u dinamičkoj memoriji delimisanom kao posabna klasa. Takve opasnosti ne prete

Listing a55

```
#include <iostream.h>
#include <string.h>
#include <alloc.h>
#include "a54.cpp"

main() {
        cout << "\nProgram A55.CPP\n";
        char * tekst1 = "Ovo je tekst1";
        Tekst S(tekst1);
        cout << "originalni tekst tekst1: " << tekst1 << "\n";
        cout << "originalni Tekst S: " << S << "\n";
        tekst1[0] = 'X';
        tekst1[1] = 'X';
        tekst1[2] = 'X';
        cout << "tekst1 posle izmene: " << tekst1 << "\n";
        cout << "S posto je tekst1 izmenjen: " << S << "\n";
}
```

Test program za klasu Tekst

Rezultat programa A55.CPP je

```
Program A55.CPP
originalni Tekst tekst1: Ovo je tekst1
originalni Tekst S: Ovo je tekst1
tekst1 posle izmene: XXX je tekst1
S posto je tekst1 izmenjen: Ovo je tekst1
```

Tekst sadrži svega nekoliko podrutinskih funkcija stvaranje i brisanje teksta odnosno ispis na ekran. U sledećoj iteraciji možemo u klasu izvedenu iz **Tekst**-a ubaciti uobičajene naredbe npr. kopiranje, poređenje, snimanje na disk itd. Pri tome se služimo preopterećivanjem operatora – i drugih. Da bismo to pravilno isprogramirali moramo se dotaći najtežeg dela C++-a – prenos argumenata i vraćanje vrednosti funkcija.

PRENOS ARGUMENATA U FUNKCIJE

Osnovna aktivnost u objektnom jeziku je stvaranje novih klasa. Glavni program ili funkcija **main()** koordinira rad objekata, a sva „pamet" programa sakrivena je u klasama. U C++ u nova klasa automatski postaje proširenje prevodioca, ali u skladu sa sintaksnim pravilima i klasama koje već postoje. Zauzvrat klasa koju napiše programer vrlo malo se razlikuje od one koju je dizajnirao C++-a predvideo kao sastavni deo jezika. Tako prevodilac proverava (statičke) tipove klasa „po broju, rodu i pridevu", automatski pripremа objekte za rad ili ih briše iz memorije, može koristiti iste operatore i za osnovne tipove jezika i za klase itd. Da bi nove klase zaista radile kao da su „oduvek" bile deo jezika, potrebno ih je pažljivo napisati: problem je u konstruktorima i operatoru = dok se prve bez naročitih problema.

Postoji tri načina da se podatak prenese funkciji – po imenu, po vrednosti i po adresi (*by neme by value by reference*). Nazovimo okvirom funkcije (*function frame*) deo memorije u kojem se nalaze sve promenljive jedne funkcije. Sintaksno tu pripadaju sva imena između vitičastih zagrada funkcije. Informaciju u tom „prostoru" menjamo

1) kopiranjem kompletne promenljive ili
2) prenosom adrese.

Prvi metod je prenošenje po vrednosti ili u prostoru funkcije pojavljuje se kopija vrednosti originalne promenljive. Kopiju možemo menjati po volji a vrednost originala se time ne menja. Drugi metod se deli na dva

a) prenošenje po imenu – ako se prenosi pokazivač odnosno
b) prenošenje adrese – ako se direktno prenosi adresa originala.

Prenos po vrednosti je sigurniji i imitira stvaranje lokalnih promenljivih funkcije kada funkcije nestaju nestaju i sve njene promenljive. Ali memorija je uvek ograničena i za jole veće strukture neracionalno je imati podatke na dva mesta ne računajući vreme potrebno za prenos.

Prenos po adresi je specijalnost C++-a i u C-u ga nema. Prenos adrese je isto što i deklarisanje argumenta kao pokazivača uz poljni prenos adrese. Prenos adresa je za programera pogodniji, prevodilac radi malo više a unutar funkcije prema takvom parametru ponašamo se kao prema imenu promenljive (nema potrebe za operatorom *). Konačno, snaga C++-a leži u dodatnim bibliotekama funkcija a prenos po adresi pojednostavljuje njihovo korišćenje.

Prenos po vrednosti ne treba odbacivati, ali on je u tehničkom smislu jednostavan samo kada su u pitanju ugrađeni tipovi C++-a. Kada prevodilac prenese **int** ili **float** dužina podatka je prevodiocu poznata, baš kao i pripremne radnje koje se moraju obaviti prilikom prenosa (postavljanje na 0, na „_" i sl.). Programer mora sam da napiše sve u vezi prenosa i vraćanje podataka za svaku novu klasu. To nije problem kada u klasi nema pokazivača i kada nije definisan konstruktor kopije (*copy constructor*). Tada prevodilac (za verziju 1.2) kopira objekat bit po bit, što je u velikoj većini situacija ispravno ponašanje. Kopiranjem „bit-po-bit" se dobija tzv. plitka kopija (*shallow copy*), jer se ignoriše bilo kakvo dodatno znanje o klasi.

Ako klasa sadrži pokazivač(e), potrebna je duboka kopija (*deep copy*) objekta. Duboko kopiranje obično podrazumeva zauzimanje slobodne memorije i kopiranje svega na sve pokazivači pokazuju. Time se osigurava integritet objekta.

SKRIVENE RADNJE KOD PREKOSA ARGUMENATA

Jedne od najvećih vrlina svakog prevodioca je što oslobađa programera od dosadnih i rutinskih operacija kao što je, na primer, prenos argumenata u i iz funkcije. C++ prevodilac, međutim, mora od programera da sazna koje sve operacije treba uraditi kada se formira novi tip ili novi objekat.

Po pozivu funkcije vrh steka se spušta nadole i oslobađa mesto za lokalne promenljive a argumenti se preuzimaju direktno sa steka. Pošto se funkcija izvrši do kraja, deo programa koji je pozvao funkciju brže argumente sa steka da bi se smanjio rizik od pozivanja funkcije sa pogrešnim brojem ili tipom argumenata (ANSI C i C++ propisuju postojanje prototipova funkcija iz kojih se već tokom prevođenja može uvideti da li se pozivi funkcije slažu pa se ovaj problem rešava na drugi način.)

Pretimo da je C prevodiocu prilično jednostavno da „guta" podatke na stek jer je broj tipova podataka mali i unapred poznat. Prenos argumenata po vrednosti je u toj situaciji vrlo lak. Prenos po imenu je još jednostavniji: samo se ne vrh steka postavi pokazivač.

Za C je normalno da se rezultati vraćaju preko registara. Pisac prevodioca sam odlučuje u kojim će se registrima vraćati rezultate za tip **int** u kojim za tip **float** i slično. Prenos preko registara je bilan da bi sa funkcija mogla zaustaviti u svakom trenutku a da se podaci ne unište. Pretpostavimo suprotno: ili da poslopi prevodilac koji vraća vrednosti preko steka. Vraćene vrednosti bi došle na slek a onda bi se vrh steka pomerio iznad adrese na kojoj je sačuvana povratna adresa za masinski **return**. Tada bi bilo moguće da se deo steka sa rezultatima obriše posle povratka iz funkcije ali pre nego pozivni deo programa uspe da obriše stek. U takvoj semi sistemsko programiranje (a ono uvek zavisi od raznih sistemskih prekida) bilo bi onemogućeno. Dakle, prenos rezultate funkcije preko registara je bilo ako C i C++ treba da budu sistemski jezici.

Šta se dešava kada je struktura podataka prevelika za registre? U C-u obično se vraćene vrednosti kopiraju u globalnu **static** oblast memorije izvan se iz funkcije pa se rezultujuća struktura kopira natrag. Takvo rešenje je opet podložno greškama posle prekida (*interrupts*). U „običnom" C-u to nije predstavljalo veliki problem jer se obično ne vraćaju strukture nego pokazivači na njih. Ali zato naredba tipa

```
A = B + C
```

gde su A, B i C strukture (**struct**) nije dozvoljena. Za ostale tipove podataka takva „složena" naredba jeste dozvoljena.

U C++-u (i nekim verzijama C-a naravno) koristi se sledeća ideja: adresa strukture (ili objekta svejedno) prenosi se „tajno" u funkciju obično preko registara. Funkcija onda kopira u vrednosti direktno u rezultat pre svog završetka. Vrlo efikasno i čisto. Međutim, to uopšte nije standardno rešenje i na njega se ne može računati ako se zeli prenosivost programa. Podaci se garantovano neće prenositi preko oblasti za **static** podatke ako je prenos preko konstruktora kopije odnosno ako je pozvan preopterećeni **operator=**.

C++ ovim „tajnim" aktivnostima dodaje i neke svoje, ali ne uvek. Ugrađeni objekti poput **int**, **float** i **char** se ponašaju isto kao u običnom C-u. Objekat je – spoja gledano – C++ prevodiocu isto što i **struct**. Ako se programer ne umeša, C++ objekte tretira potpuno isto kao i **struct**. Međutim, klasa je mnogo više od strukture: uvek postoje neka pripremne, brisanje, pokazivači na druge objekte itd. Zbog svega toga prosto kopiranje strukture nije dovoljno dobro za klase. C++ kroz pojam konstruktora kopije (*copy constructor*) dozvoljava programeru da se „ličko" umeša i sam definiše šta će se dogoditi tokom prenosa podataka u funkciju i van nje.

KONSTRUKTOR KOPIJE X(X&)

Konstruktor kopije je konstruktor čiji je argument adresa objekta iz iste klase. Obično se koristi za pocetno postavljanje i naravno, za kopiranje podataka. Konstruktor kopije se za početno postavljanje koristi kada želimo da deklarišemo i poslavimo objekat na osnovu već postojećeg objekta iste klase (a56.cpp).

Listing a56

```
//          konstruktora kopije
#include <stdio.h>  // ako ne puts()
class kk {
        int i;
public:
        kk(int j) { i = j; }
        // konstruktor kopije
        kk(kk & vv) {
          puts("poziv konstruktora kopije");
          i = vv.i;
        }
};
main() {
        puts("\nProgram A56.CPP");
        kk original(1);
        kk copy1(original); // koristi se konstruktor kopije
        kk copy2 = original; // ali i ovako!
}
```

Pocetno postavljenje pomocu konstruktora kopije

Rezultat ovog programa iznenađuje – konstruktor kopije **copy1** pozvan je dva puta!

```
Program A56.CPP
poziv konstruktora kopije
poziv konstruktora kopije
```

Za prvi poziv je jasno jer u programu piše

kk copy1(original),

Drugi poziv poliče od sledećeg reda i veoma je nelogičan. Naviknli smo da znak = znači pridodeljivanje a u ovom – i samo u ovom – slucaju operator = nikad ne biva pozvan vec se izvrsava konstruktor kopije. Srecom, oblik

ImeKlase objekta = objekat

je izuzetak i kao takvog ga moramo pamtiti. Kao da to sve nije dovoljno u slučaju deklaracije

```
ImeObjekta objekat1,
ImeObjekta objeket2,
objekat1 = objekat2
```

konstruktor kopije takođe ne biva pozvan vec se izvršava **operator**=() (pod uslovom da je definisan)

KOPIRANJE REZULTATA

Druga uloga konstruktora kopije je da prenosi parametre po vrednosti i da vraca rezultate iz funkcija. Ako konstruktor kopije postoji prevodilac ga automatski poziva kadgod je objekat prenosi ili vraća po vrednosti. Smisao tog pravila je da postoji sistemski definisano mesto za prijem i vracanje podataka. Kod prenosa u funkciju lokalni objekat funkcije mora popnimiti vrednosti objekta iz okruženja. Obratno, kod predaje rezultata spoljnji objekat prima vrednosti lokalnog objekta. Pripreme objekta se najčešce svode na kop-ranje ali moze obuhvatiti i mnoge druge radnje. Kao primer navedimo nastavak klase **Tekst** nazovimo je **ObicanTekst**. Evo programe A57.CPP

Listing a57

```
#include <iostream.h>
#ifndef OBICANTEKST_H
#define OBICANTEKST_H
#include <string.h>
#include "a54.cpp"
class ObicanTekst : public Tekst {
public:
        ObicanTekst::ObicanTekst(char * t) : Tekst(t) { }
        ObicanTekst::ObicanTekst(ObicanTekst & tekst) {
          tkst = new char[strlen(tekst.tkst) + 1];
          strcpy(tkst, tekst.tkst);
          }
        ObicanTekst operator=(ObicanTekst & tekst) {
          delete tkst; // oslobadja zauzetu memoriju
          tkst = new char[strlen(tekst.tkst) + 1];
          strcpy(tkst, tekst.tkst);
          return *this;
        }
};
#endif OBICANTEKST_H
main() {
        cout << endl << "Program A57.CPP " << endl;
        char * t = "obican tekst";
        char * ut = "novi tekst";
        ObicanTekst ot(t), jt(t);
        jt = ut;  // i ovde konstruktor kopije!
        cout << "ot" << ot << endl;
        cout << "ut" << ut << endl;
        cout << "jt" << jt << endl;
        ObicanTekst nt("tekst za et");
        cout << "nt" << nt << endl;
        ObicanTekst wt(et);
        cout << "wt" << wt << endl;
        ObicanTekst qt = et;
        cout << "qt" << qt << endl;
}
```

Klasa ObičanTekst

Rezultat ovog programa je

```
Program A57.CPP          el tekst za et
ot obican tekst          wt tekst za el
nt novi tekst            qt novi tekst
jt novi tekst
```

Konstruktor kopije prepoznajemo po prisustvu adresnog operatora &. Ovde bez njega nismo mogli jer bi memorija za **tekst** ostala van domasaja u slučaju obicnog kopiranja (koje sa izvršava automatski kad konstruktor kopije nije naveden)

Konstruktor kopije se implicitno koristi u sledećim slučajevima prenosa po vrednosti u prvom redu u funkciju u drugom redu iz funkcije

```
void nesto(X c) {   // prenos po vrednosti, poziva se X(X&)
return c;           // vracanje po vrednosti, poziva se X(X&)
}
```

OPERATOR =

Operator = je poseban slučaj. Za njega ne važe pravila kao i za ostale operatore. Ne možemo ga nasleđivati automatski iz klase u klasu. Najkrace ga mozemo opisati kao destruktor plus konstruktor kopije. Medutim, konstruktor kopije je pre svega konstruktor a **operator** – to nikad nije. Svaki konstruktor vraca vrednost, ali to je uvek **return *this**. **operator**= može vratiti bilo šta a naravno najčesci slucaj je da vraća vec spomenuti **return *this**. Zahvaljujuci tome u C++ u je moguce pisati izraze tipa **A=B=C** za klase.

Operator = u klasi **ObicanTekst** verno sledi sva ova pravila: prvi red je identican destruktoru iz osnovne klase **Tekst** sledece dve linije su prenesene iz konstruktora kopije klase **ObicanTekst** i na kraju vracamo celu strukturu pomoću ubičajenog **return *this**.

C++ dozvoljava programeru da „dovrši" objekat tj. da preuzme potpunu kontrolu nad svim detaljima. Konstruktor kopije je najbolje izbegavati ako je moguće. Da bi to sto češće bilo moguce u verzji 2.0 u odsustvu eksplicitnog konstruktore kopije podrazumeve se da prevodilac provodi prpiremu član po član tj. da poziva konstruktore kopija za svaki clan klase. Ako član klase sam sadrzi pridruzene objekte poziva se i njihovi konstruktori kopija itd. Isto važi i za operator = . To je mnogo sigurnije nego kopiranje objekta bit po bit. Iako da u verzji 2.0 otpada veliki deo problema koje prv C++ programeri imali. No ako programer zaželi da sam napiše konstruktor kopije ili operator = i dalje mu – s jedne strane – ništa ne stoji na putu ali – s druge strane – sam preuzima odgovornost za rad prevodioca.

KAKO PISATI KLASE I HIJERARHIJE

Dopisimo sada klasu **ObicanTekst** do nivoa upotrebljivosti. Pošto je najteži deo – zauzimanje memorije – upravo napisan ostaje nam I | dizajniranje klase i 2) tehnička realizacija. Pod dizajniranjem klase podrazumeva se i izbor operacija koje želimo na ovom nivou da uvedemo ali i izbor operatora i imena funkcija kojima te operacije pridodeljujemo. Pojam leksika ocigledno zasluzuje finu hijerarhiju. Na najvišem nivou to je klasa **Tekst** odmah ispod nje je **ObicanTeksi** tj. klasa koji bi trebalo da sistematizuje sve čitanje operacije nad tekstom a dalje mozemo razviti posebnu klasu za rad sa leksikom na PC racunarima. Iz nje možemo nasleđivati dve klase jednu za rad u tekstuelnom rezimu ekrana i drugu za grafiku. Broj klasa je ocigledno veliki i zadatak dizajnera hijerarhije je da na svakom nivou uspostavi umerenu funkcionalnost svake klasa treba da bude sposobna da donekle i sama obradi podatke ali i da postavi osnovu za daljnje nasleđivanje. Do dobrog rasporeda operacija u jednoj hijerarhiji klasa oblazi tek posle duzeg vremena i iskustva.

Uočavanje objekata počinje *pre* pisanja programa. Naravno programer može sesti i odmah početi sa ukucavanjem naredbi nadajuć se da će vremeno „sleci osećaj" za problematiku. Ipak objektno programiranje počinje objektnom analizom i objektnim dizajnom. Pod objektnom analizom podrazumevamo uočavanje objekata u sistemu. Rezultat analize sistema je objekat samo ako „pod istu kapu" poslavimo i operacije i podatke. Praktično bilo šta može biti objekat ali neka pravila ipak postoje. Najlakše kao objekte prepoznajemo fizičke objekte poput slike kuće sveske ekrana, misa i sl. Objekat može i neki algoritam ali i neka veza zmedu drugih objekata. Najamnina video kasete je primer nemalen jalnog ali svakako važnog objekta u poslovanju videoteke. Za objekte obično uzimamo i načine odnosa između već ustanovljenih objekata. blok dijagrami grafički prikazi struktura podalaka i slicno.

Po završenoj objektnoj analizi počinjemo objektni dizajn programa tj. na nivou objekata modeliramo rešenje problema. Ako odabrani objekti mogu da sarađuju lako da se problem moze rešiti, prelazimo na (pažljivo) kodiranje na prigodnom jeziku i program je gotov! Objektne analiza i dizajn ne zavise od objektnih mogucnosti (ili nemogucnosti) konkretnog jezika ali će kodiranje biti mnogo lakše ako jezik ne sintaksnom nivou podržava klase.

Proces objektne analize i objektnog dizajna ima smisla ponavljati i tokom pisanja programa jer probleme iz neračunarskog sveta prevodimo u računarske objekte popul diskova. Iraka imenika miševa ekrane tastature indeksa računarskih mreža i mnogih drugih „tehnički nastrojenih" pojmova. Programiranje bez objekata zahteva da programer radi više tj. da detaljnije kodira. Ako je jezik objektnoorijenisan i ako postoje programske realizacije računarskih resursa (npr. objekat **Tekst** ze pisanje teksta objekat **Ekren** za pisanje po ekranu itd.) programer će na visem nivou završiti kodiranje pa će ceo postupak programiranje biti pre gotov.

Kada dođemo do prve liste objekata tj. do kandidata za objekte u našem novom programu mozemo početi sa izvođenjem novih klasa na osnovu le liste. Obično trazimo da se nove klase mogu nasleđivati da smanjuju intenzitet transfera podataka između modula ili prosto da sadrže veze između drugih objekata. Vazi pravilo da se isla informacija ne sme naci u dva objekta. Ako treba povezati dve klase najbolje je osnovali trecu klasu sa iskljućivim ciljem da kodirane prve dve.

Česi izvor objekata su apstraktni tipovi podataka. Tu sa obično misli na standardne strukture kao sto su redovi čekanja, stek, skupovi, liste, slabla. zaslo ih programirali iznova ako se mogu lagodno nasleđiti? Za C++ se vec nude hijerarhije klasa za ovakve hijerarhije kompletna finansijska matematika kompletan matricni racun, konsnicki interfejs i štosla drugo. Takve objekte programer treba da izuči i onda samo nasleđuje uz eventualne male modifikacije.

Multitasking se prirodno identifikuje sa objektima kao u nekoj *fantasy* igri svaki objekat020postoji nezavisno od svih drugih a glavni program ih samo koordinira u (simulacionom) vremenu. Zapravo je objektno programiranje najsličnije pisanju simulacija.

Naravno najlakše je u objekte pretociti vec postojeću hijerarhiju iz neke

naučne oblasti – biologije istorije medicine matematike Prisustvo hijerarhije najlakše otkrivamo relacijom „pojam A je isto što i pojam B uz male izmene" Čovek je sisar pa bi klasu **Sisar** nasleđivala klasa **Čovek** a iz ove bi mogli generisati klase **Muškarac** i **Žena** itd Znanje počinje klasifikacijom a objektno programiranje je samo jedan formalni računaru prilagođen izraz te činjenice

Sam način stvaranja hijerarhije može biti i *top-dole* i *bottom up* Ij mozemo je dedukovati misaono ili indukovati istrazivanjem Članove iste hijerarhije prepoznajemo pitanjima „Šta je zajednicko ovim potencijalnim objektima?" ili „Koja klasa bi mogla biti osnovna za ovaj objekat?" Međutim bez obzira kako dođemo do neke hijerarhije uvek je dokumentujemo u *top-dole* maniru Ij počinjemo od najopstijeg objekta i završavamo sa najkonkretnijim

Slicno od velike koristi je i uklanjanje objekta iz poslojeca hijerarhije– mozda ge mozemo podvesti pod neki vec poznat objekat

DOVRŠAVANJE OBJEKATA

Gomji knlenjumi za stvaranje objekata su pandan poznatim zahtevima iz oblasti softvarskog inzenjeringa (*software engineering*) Za objekte vaze i neki drugi njima prirodeni kriterijumi Jednim imenom ih možemo nazvati „teznjom za kompletiranjem objekta" Kod prvog „upoznavanja" sa novim objektom još ne znamo sve o njemu ali uvek raspolazemo mi n mumom operacija i internih podataka Te operacije se uglavnom odnose na problem koji rešavamo (nace ih se ne bi ni sešli nti bi nam baš taj objekat bio potreban) Minimalni skup operacija u objektu moze biti dovoljan da program proradi ali konsnik objekta uvek očekuje i nesto vise ili pi rode probleme ukazuje na neminovnost promene u faz eksploa laoje sistema Zato treba težiti ka kompletiranju objekta. Evo nekoliko kriterijuma za to

Za svaku operaciju treba obezbediti i inverznu operaciju. Najcešci oblik je *undo* Ij ponistavanje efekta prethodne radnje Svako od nas bi bio gidno razočaran da procesor020leksta ili editor sa kojim piše nema lakvu mogucnost pa zasto to ne bi ugradivali i u programe koje sami pisemo i prodajemo drugima!? Osim toga de b se operacija *undo* kvalitetno realizovala obicno treba pretresti i samu osnove klase pa je to dobar nacin da otkrijemo propuste u objektnoj analizi

Druga vrsta inverzna operacije je suprotna operacija ako jo ima smisla ze datu situaciju konsnik klase ce ranije ili kasnije zatražiti i suprotnu radnju (koje može biti ponistavanje ali i ne mora)

Posle inverznih dodajemo komplementarne radnje One prošliuju ideju suprotne radnje os m što kod ove posloji samo jedan suprotni element dok u principu može postojati mnoštvo lakvih „suprotnih" elemenata. Evo spiska suprotnih i obralnih radnji Primetimo da se tu za svaku radnju A ne navodi obavezno *non* A

suprotne radnje

širenje	skupljanje
deobe	kombinovanje
napredak	povlačenje
ulevo	udesno
dodaj	oduzmi
unapred	unazad
povecanje brojača	smanjenje brojača

komplementalne radnje

dodaj element	izmeni element
dodaj element	broj elemenala
unepred	prelirazivanje od počatka
ulevo	udesno pravo unezad
počelak blojanja	prekid brojanja

Zatim logicki kompletiramo skup operacija u objektu To može biti vrlo komplikovano jer se trudimo da predvidimo sve sta se može zahtevati od objekta, naročito u budućnosti Jedan nacin je da dodamo sve „moguće i nemoguce" operacije a drugi je da dopišemo samo one koje će verovatno biti potrebne Ovaj drugi prilaz je naravno preporučljiv

Primetimo da je ovo i dalje samo laza kreiranje objekta te da ovde još nema govora o konkretnzaciji objekta u nekom jeziku ili radnoj sredini Objektni dizajn treba (i može) da bude nezavisan od konkretnih detalja ali svakako doci ce cas kada dizajn treba da postane deo programe Ne moramo isprogramirati sve što smo u objektu predvideli ali vazno je osmisliti kako bi to jednog dana mogli *bezbolno* uraditi

NASTAVAK KLASE ObicanTekst

Koncipiramo klasu **ObicanTeket** u skladu sa gornjim idejama Tu klasu pi šemo da bi sistematizovali posljepce operacije sa lekstom Pocnimo od funkcija iz sistemskog zaglavlja STRING H Tamo su nabrojane funkcije obicnog C-a popul **strcat** za spajanje dva leksta **strcmp** za poređenje **strcpy** za kopiranje itd Ispisimo ih u objektnom maniru npr pomoću preopterecenih operatore U sledecoj tabeli su operatori i odgovarajuce funkcije

operator	funkcija
+	strcal
==	poređenje na jednakost
!=	najednakost
>	strikino veće
<	strikino manje

Evo dela daloteke A58a CPP u kojoj je klasa **ObicanTekst** donekle dopisa na u skladu sa ovim željama

Listing 58.cpp

```
                                *
        ObicanTekst ObicanTekst::operator+(ObicanTekst & tekst) {
          strcat(tkst, tekst.tkst);
          return *this;
        }
        int ObicanTekst::operator==(ObicanTekst & tekst) {
          return !strcmp(tkst, tekst.tkst)==0;
        }

        int ObicanTekst::operator!=(ObicanTekst & tekst) {
          return !strcmp(tkst, tekst.tkst)!=0;
        }
        int ObicanTekst::operator<(ObicanTekst & tekst) {
          return !strcmp(tkst, tekst.tkst)<0;
        }
        int ObicanTekst::operator>(ObicanTekst & tekst) {
          return !strcmp(tkst, tekst.tkst)>0;
        }
};
#endif OBICANTEXST_H
main() {
        cout << endl << "Program A58.CPP " << endl;
        char * t = "obican tekst";
        char * nt = "novi tekst";
        char * rt = "treci tekst";
        ObicanTeket ot(t), jt(nt), qt(rt);
        cout << ot < jt << endl;
        cout << (rt == jt) << endl;
        cout << (qt == jt) << endl;
        cout << (ot != jt) << endl;
        cout << (qt < jt)  << endl;
        cout << (qt > jt)  << endl;
}
```

Modifikacija klase ObicanTekst

mo dogovoriti da *undo* bude direktna suprotnost od operatora = Ij od postavljenja vrednosti teksta Naša klasa **ObicanTekst** treba dakle da bude u stanju da zapamti prathodnu vrednost interne promenljive **tkst** i da ga vrati po potrebi To implicira posljenje „rezervne" interne promenljive nazovimo je **rezTkst** Tu promanjivu mozemo deklarisati kao internu za celu klasu ali onda ce i svaki primerak klase **ObicanTekst** i svaki primerak svake klase izvedene iz **ObicnogTeksta** da sadrzi i **rezTkst**. Ustedeli bismo na memoriji ako bi **rezTkst** bio globalne statička promenljiva– i takve narzgled naivne ali zapravo dalekosezne odluke zavise od dizajnera!

Možde bi bilo dobro izdvojiti novu klasu **UndoTekst** razmišljajući ovako: **UndoTekst** je isto što i klesa **ObicanTekst** ali sadrzi dodatnu internu promenljivu **rezTkst** i odgovarajuću priduzenu funkciju **undo**

Dalje kao suprotne operacije mozemo uvesti npr **operator-** koji bi liče na razliku skupova i teko bio suprotan ali ne i jedinstveno invetzan za **operator** Naročito je teko uvoditi suprotna poređenja i sortiranja Za potrebe ovog leksta ne moramo ubaciti sve supiotne i obiatne radnje ali ih mozemo naznačiti operacija **ulevo** za leksl mogla bi biti kruzno pomeranje leksta (*levo shift*) ulevo a suprotne operacija **udesno** je kruzno pomeranje udesno (*desno shift*) Operacije **unapred** i **unazad** bi – analogno operacijama ned bitovime u bajtu – mogle biti pomeranje ulevo i udesno ali bez kruznog prabacivanja znakova sa kraja na počelak leksta i obralno

Komplementarne radnje (između ostalog) mogu biti prelrazivanje unepred od pocelke od zadatog slova u leksiu zatim isto to samo unatrag Slicno je za sortiranje – po ASCII vrednosti znakova alfabetsko abecedno po prvom slovu ako se leksi sastoji iz reci i sl

Na nivou C++-a ovim operacijama možemo pridodeHi operelore i lo jedan ili vise (opel preopterecivanje!)

Kada smo iscrpli sve preporuke preostaje nam još jedna pustli masli na volju i nobiejel sve moguce operacije U našem slucaju ocigledan je nedostatak ulazno-izlaznih operacija u klasi **ObicanTekst** To je nemerno jer ću klasu ce nasleditl (ako je budemo napisali) klasa **PCTeket** ne UNIX mašinama mogla bi je nasleditl klasa **UNIXTeksat** – buduca klase **WindowsTeksat** i **OS2Tekst** da i ne spominjemo!

PERJANICA OBJEKTNOG PROGRAMIRANJA

Na najnizem nivou. C++ je „bolji" C Strozije je ali lo je samo u konst programera Brzina izvršavanja nije manja nego sa obicnim C'om pa je i sa le strane dobar kandidat za pisanje komercijalnih programa Veoma ja fleksibilan i snaga mu rasle sa brojem dodatih hijerarhija klasa.

Objektna proširenja su u C++-u jednostavna sintaksa im je svedena na minimum i ne zahteva mnogo učenja Miliont profesionalnih programera ionako vec znaju 80% C++-a! Zalo je on u velikoj ekspanziji i sve su sanse da ce na prirodan nacin istisnuti C Ime ga i pod UNIX om i pod DOS om i pod *Windowsima* C++ je perjanica020objektnog programiranja – nacin i sredstvo da prednosti objektnog programiranja zažive u nasim računarima.

Zoran Životi

++ i SPOREDNI EFEKTI

C kao malo koj drugi jezik daje prilično slobodu programeru u formiranju složenih izraza. Bezbroj puta smo se oduševljavali kako se čitave linije koda mogu sažati samo u jednu jedinu i kako čak i tada postoja mogućnosti za još gušće pisanje i još kraće izraze. Ovakva sloboda mora imati i svoju cenu. U C u cena se zove „sporedni efekti“ i često može biti visoka.

Operator ++ kojim se vrednost promenjive povećava za jedan je odličan primer kojim se ovo može ilustrovati. Kao što je poznato od položaja operatora zavisi vreme njegovog dejstva – ukoliko se nađe ispred promenjive vrednost promenjive se menja pre nego što se upotrebi, ako je iza njegovo dejstvo stupa na snagu posle upotrebe vrednosti promenjive. Odmah da razjasnim i neke nedoumice kojima sam i ja delimično kumovao. Svojevremeno sam u dodatku „Računara“ o C jeziku kod objašnjenja aritmetičkih operatora pisao da se vrednost promenjive ne menja nakon što je ona upotrebljena na svim delovima izraza. To bi konkretno značilo da nakon sledeće linije

```
i = 1,
j = ( k = i++ ) + ( i = i++ ),
```

promenjive imaju sledeće vrednosti

```
i = 3,
j = 2,
k = 1,
i = 1,
```

Iako bi ovakvo dejstvo ++ operatora moglo da bude poželjno (i sa mnogo manja sporednih efekata!) ono na žalost nije tačno. Definicija koja je data u bibliji C-a (Kernighan & Ritchie *The C Programing Language*) je doslovno

Izraz i++ povećava vrednost promenjive n pre nego što se upotrebi njena vrednost. n++ povećava vrednost nakon što je vrednost upotrebljena.

Ovo konkretno znači da je u prethodnom izrazu redosled dešavanja sledeći

```
k dobija vrednost originalnog sadržaja i (k = 1)
i sa povećava za jedan (i = 2)
i dobija vrednost trenutnog stanja i (j = 2)
i sa povećava za jedan (i = 3)
j je suma k+i (j=3)
```

I tu nastaje problem! C po definiciji ne garantuje nikakav redosled izvršavanja izraza naravno pod uslovom da su operacije istog prioriteta. Tako se može desiti da se dio izraza i = i++ izvrši pre dodele vrednosti k promenjivoj – iako će ceo zbir kao i končan zadržati i promenjivu bih isti k i i će imati drugačije vrednosti. Potpuno ista stvar se može desiti i u sledećem slučaju

```
printf( "%d %d", k=i++, i = i++),
```

jer se ni redosled prosleđivanja parametara ne garantuje. Sa redoslednom prosleđivanja parametara se može nekako i izaći na kraj – radi se o mašinski zavisnoj operaciji i kada jednom ustanovite kojim redosledom to radi kompajler u vašem okruženju ne njega se da lje možete oslonit. Kod računanja izraza se takođe može uhvatiti neka slaba veza, ali treba imati na umu da ovde može da se umeša i optimizator koda.

Ipak praksa u kojoj sa ostanjale na ovakve stvari je više nego štetna. Time rušite jednu od najlepših osobina C a – prenosivost na različite platforme i za pravo u program polpuno nepotrebno slavljate neku vrstu tempirane bombe koja je vrlo lesko otkriti. Sporedni efekti se pojavljuju uglavnom u sledećim slučajevima

a) višestruka upotreba ++ opetatora
b) dodela vrednosti promenjivoj unutar složenog izraza
c) prosleđivanje parametara funkcije u obliku izraza

Takođe obratite pazn ju i kod konstruisanja i upotrebe makroa jer oni u sebi mogu da kr ju upravo ove operacije koje mogu izazvati sporedne efekte.

Dakle iako vas C prosto vuče da skrolite sebi posao pisanjem na primer

```
a[i] = i++;
```

nemojte to nikada da radite

TVRDOGLAVA FUNKCIJA

Da funkcija iz C biblioteke često ima neko neočekivano ponašanje (iako sasvim logično) već smo vi dali na primeru **strncpy** funkcije. Evo još jednog slučaja sa kojim možete da se pogrešite

Varijam da funkciji FOPEN na traba nikakvo uvodno objašnjenje jer je više nego često korišćena. Pored parametra – naziva datoteke – tu je i drugi string parametar kojim se definiše režim u kome se datoteka otvara. Konsta se znaci **r (read)** **w (write)** **a (append)** + dva opciona **b** (binarni režim) i **t** (tekst rezim). Ova slova se mogu kor sti u nekoliko kombinacija. Pre standardizacija C-a bilo je uobičajena po da se koristi **wr** i **rw** kombinacije kao oznaka da se datoteka otvara i za čitanje i za pisanje, ali je sada po ANSI standardu situacija preb rena u principu „jedno slovo i (opciono) +“. Tako „r“ otvara datoteku samo za čitanje r+“ i za čitanje za pisanje s tim što datoteka mora da postoji. Slično tome „w“ kreira (ili uništava sadržaj postojeće) novu datoteku i podrazumeva da će u nju samo biti pisano. Dodatkom „w+“ datoteka postaje otvorena i za čitanje i za pisanje.

Najinteresantniji parametar je „a“ (**append** dodaj na kraj). Pretpostavimo da program traba da ispisuje leksi u datoteku (naka se radi o opcija „print to file“). Uobičajeno je da se ispis stalno slaže u istu datoteku. Da ne postoji opcija „a“ procedure otvaranja ovakve datoteke bi mogla bi sledeća

```
if ( (f=fopen(filename,"r+"))          NULL )
    fopen(filename,"w");
else
  fseek(f,0L SEEK_END)
```

Pošto režim „w“ uvek poništava sadržaj datoteke prethodnim „r+“ smo pokušali da ustanovimo da li da toteka vec postoji da bi se tada poslavili na njen kraj i time dodali novi ispis. Opcija „a“ ovo poslize u jednom koraku. Nakon

```
f = fopen(filename,"a");
```

imamo ili novu datoteku (**fopen** je kreira) ili otvorenu vec postojeću datoteku sa pozicionim na njen kraj. „Zabava“ počinje ako često dodavanje sadržaja nije ono što ste nameravali. Pretpostevimo da postoji datoteka dužina 6 bajtova sa sledećim sadržajem

```
223456
```

Datoteku otvorila opcijom **append** zelim sledi

```
fprintf(f,"%c", 7);
```

i datoteka dobija sadržaj 2234567. Zatim se pojavi potreba da onaj prv bajt (2) promenite u 1. Logično bi bilo pokušati

```
fseek(f,0L SEEK_SET),
fprintf(f,"%c", 1 );
```

ali vas nakon pregleda sadržaja datoteke čeka malo iznenađenje

```
22345671
```

Kada jednom otvorite datoteku na ovaj način svako zapisivanje uvek ide na kraj datoteke bez obzira što se inače sasvim legalno menjali poziciju file pointera **fseek** je kod ovog režime dozvoljen iz sasvim drugih razloga – ako upotrebite opciju „a+“ dlato leka će moći da se čita i lada će pomenuta funkcijom **fseek** imati smisla

GETENV

Za prenošenje parametara iz okruženja u C program postoji nekoliko mehanizama o kojima je ovde bilo reči. Pisao sam o jednostavnosti kojom se para metri (zahvaljujući **scanf** grupi funkcija) mogu prosleditи preko konfiguracione datoteke kao i direktno putem komandne linije. Ipak postoji odreden tip parametara za koji je treći do sada nepomenuti način prenosa parametara daleko najpogodniji. Radi se o prenosu putem **environment** promenjivih. PATH i slične promenjive su deo sistemskog okruženja, ali sa u isto područje može kao što je poznato, ubaciti bilo kakav nazivni parametar i njegova vrednost sva dok se može zadestavi u obliku „IME_VREDNOST“. Parametri ovog tipa bi se po svojoj suštini najbolje mogli kalifikovati kao „opcioni i nepromenjivi“. Evo jednog konkretnog primera

Često programi imaju potrebu za privremenim radnim prostorom na disku. Bilo jem lekućeg diska/direktorija nije dobro resenje jer korisnici najčešće imaju na raspolaganju neku particiju koju progla šava ju kao „radnu“ i na kojoj je više prostora nego ikamo gde su programi. „gusto“ složeni u neku logičku celinu. Kompilirovan program pretpostavljanjem svih raspoloživih diskova u pokušaju da slobodnim prostorom je zaista napotrebno u situaciji kada korisn k ima jednostavan način da programu lo sam po svom izboru saopšti. Dovoljno je da u svoj **autoexec** doda liniju poput

```
SDRTEMP=C:\TEMP
```

C programu ostaje da ovu vrednost konsultuje što se zvodi izuzetno lako zahvaljujući funkciji **getenv**. Njena deklaracija je

```
char *getenv( char *vername );
```

i vraća pointer na vrednost ukoliko postoji **varname**. U našem primeru izbor promenanog radnog direktorijuma možamo obavi ti na sledeci način

```
#include <stdlib.h>
char workdir[70]
void setworkdir(void)
{
  char *val,
  if ( ( val = getenv( SDRTEMP ) ) != NULL )
     strcpy( workdir, val );
  else
     strcpy( workdir, "\\" );
}
```

Treba nagla ti da pointar **val** pokazuje direktno na samo **environ** područje koje nja preporučuje ma njali direktno. To konkretno znači ako imamo potrebu da na ovako prauzel direktorij dodamo naziv neka **temp** datoteke onda je obavezno kao što je to u primeru urađeno kopiral vrednost u neku promenjivu u samom programu. Ne zaboravite takode da uključite **stdlib** zaglavlja. Ili semi da deklarišete da **getenv** vra ća **char ***.

FP_SEG i FP_OFF

O „mastu i ulozi pointera u životu C programera sam do sada vec više puta pisao. Bez obzira na kol či nu vrantarna potrošenog ne njih uvek će se nac još neki primer koji zavređuje paznju. Evo dakle jos jedne „a dresne gimnastike“ koja je u C u sasvim legalna.

Direktna dodala vrednosti pointeru se najčešće obavlja uz pomoc **cast** operatora i ini li **long** promenjiva. Video memoriji se moze pristupit pointerom ko me prethodno dodelimo sadržaj ne sledec nacin

```
unsigned far *videomam    (unsigned far *)
0xb0000000L
```

Bez potrebe da dokazujam zašto nje dobro na vodim samo i mogucnosi da se sta stvari obavl bez upotrebe pointer varijable – dovoljno je da direktno pišamo

```
*( (unsigned far *)0xb0000000L ) = 0x7020;
```

Medutim ponekad je zgodn je far po ntaru dodeljivati vrednost segmenta ofseta odvojeno. Tada se možemo poslužiti makroima koji su definisani u MSC dos h zaglavlju

```
FP_SEG( videomem ) = 0xb000
FP_OFF( videomem ) = 0
```

Šta se koja iza ovih makroa? Evo kako izgleda njihova definicija

```
#define FP_SEG(fp) ( (unsigned far *)&(fp) + 1))
#dalina FP_OFF(fp) ( (unsigned far *)&(fp)))
```

Iako deluje tipicno pinterski i C-ovsk zamišeno suština je jednostavna. Ako je fp neki pointer onda se **edress off** operatorom saznaje njegova adresa potom **cast** operatorom ova adrese tretira kao pointer na naku **unsigned** promenjivu. Raširenjeno na laza upotrebom pomocna varijable procedura bi izgledala ovako

```
unsigned far *ofs
ofs = (unsigned far *)&fp
*(ofs + 1) = ...
```

Ovo naravno nije jedini način da se obavi posao. Možemo kor sti i pratvaranje pointera u **long** proma njivu pa zatim ar operatorom „ugla vi“ olsetni i segmentn deo (uz pisthodni **shift**). Ipak prv oblik do zvoljava kompajeru da ga idealno optimizuje – citava ona slozena konstrukcija se na kraju prevodi u samo jednu masinsku instrukciju za direktnu dodelu sadržaja nekoj memorijskoj lokaciji

Ono što je kod personalnih računara dobro za korisnika a loše za programere jeste da do puštaju ugradnju kako se to kaže širokog spektra najrazličitijih grafičkih adaptera Od dobrog softvera se očekuje da podrži ako na svaki a ono barem tri-četiri najrasprostranjenija standarda. *Turbo Pascal* nam omogućava da svoje programe učinimo kompatibilnim sa dosta grafičkih kartica (iz priložene tabele), a i vi še ako posedujemo pripadajuće drajvere

INICIRANJE GRAFIKE

DetectGraph (Gd,Gm) procedura predstavlja prvi korak pri dizanju grafike Vi njoj slavljate na raspolaganje varijable za **GraphDriver** i **GraphMode** inicijalizuje lipa a ona vam ih napuni sa određenim vrednostima u zavisnosti od vrste grafičke kartice instalirane na računar Ako konstatuje da ona posjeduje **Gd** će primiti neku vrednost iz druge kolone tablice u suprotnom će vratiti -2 **Gm** će sadržavati vrednost za grafički mod najveće rezolucije te grafička kartice (četvrta kolona)

Sada prosleđujete **Gd** i **Gm** proceduri **InitGraph (Gd,Gm,Path)** Na njoj je da nas prebaci u grafički režim koristeći određeni **BGI** drajver Ako se on ne nalazi u tekućem direktorijumu onda treba u string varijabli **Path** navesti put do njega

Često se ne diskete koju ste dobili uz grafičku karticu nalazi i proizvođačev BGI drajver Funkcija **InstallUserDriver (Naziv, Pointer)** omogućava njegovo uključivanje u vaš program Funkciji prenosite dajdaleko naziv drajvera bez ekstenzija i eventualno pokazivač na sopstvenu funkciju za autodetekciju (umesto **Detect**) Koga zanimaju detalji u vezi ovoga može proučiti primer korišćenja VGA256 BGI iz pretprošlog broja

Evo jedne nove rubrike na ovoj strani ćemo svakog meseca donositi kratke priloge vezane za programski jezik paskal Tačnije za Borlandov *Turod Pascal* – naki drugi kompajler koji bi mu mogao konkurisati još nije na vidiku Služićemo se verzijama 5 0 i 5 5 nećemo objašnjavati poznate i trivijalna stvari, koje se mogu naći u samim priručnicima ali se jakoće nećemo bar ne u početku, upuštati ni u objektno programiranje u koje nas uvodi verzija 6 0 Prvo izdanje rubrike donosi neke malo prostije stvari vezane za grafiku a u narednim ćemo se baviti nešto složenijim temama Objasnićemo kako sa *Turod Pascal* povezuje sa drugim jezicima (prvenstveno sa asemblerom) kako se formiraju sopstvene biblioteke na koje se sva načine mogu prenositi argumenti funkcijama i procedurama, gde se koristi rekurzija i još dosta toga

CloseGraph procedura treba da završava svaki prevedeni program koji koristi grafiku Inače se može desiti da se vratite u DOS ali ne i u leksi-mod!

PROPORCIJE EKRANA I BIT-MAPE

Sigurno ste primetili da u Herculesovom grafičkom modu iscrtani krugovi na liče baš puno na krugove. vec više na elipse Uopšte sva figure na prikazu su pomalo „spljoštene", kao kada gledate bioskopski film na TV-u Uzrok tome je nesrazmera između odnosa dužina i visine ekrana monitora i bit mape kartice Naime većina monitora ima taj odnos približno 3 2 dok je horizontalna rezolucija Herculesa 720 a vertikalna 348 Jačkica što dođe nešto više od 2 1 Stvari stoje još gore kod CGAHi sa 640×200 piksela – odnos je tačno 3 2 1 Odmah se nameće pitanje zašto su se njihovi projektanti opredelili baš za takva rezolucije ako su znali da će to dovesti do vrlo primetnih deformacija slike? Ima dva razloga .

Prvi je taj što su oni hteli pošto-poto da stvore kartice sa što više različitih modova A to se najlakše postiglo jako što se video čipu saopšti koliko susednih oktova video memorije da uzima u obzir pri određivanju nijansa jedne tačkica na ekranu Varirali bi dakle broj boja ali i horizontalna rezolucija dok bi vertikalna ostajala ista Tako bi imali jedan ili više modova (kod CGA kartice to su oni 320×200) gde je rezolucija niza ali sa nekoliko boja i jedan monohromatski namanjen pre svega za ispis teksta

Drugi razlog leži u činjenici da je sa stanovišta konstrukcije katodnih cevi daleko lakše povećavati horizontalnu nego vertikalnu rezoluciju monitora Mlaz elektrona kao što je poznato velikom brzinom osvetljava (ili ne osvetljava) fosforescentni premaz na prednjoj strani cevi iduci prvo horizontalno od početka do kra-

Slika 1 GRAFIČKI MODOVI

Kartica	Gd	Naziv moda	Gm	Rezolucija	Boja
CGA	1	CGAC0	0	320x200	4*
		CGAC1	1	320x200	4*
		CGAC2	2	320x200	4*
		CGAC3	3	320x200	4*
		CGAHi	4	640x200	2
MCGA	2	MCGAC0	0	320x200	4*
		MCGAC1	1	320x200	4*
		MCGAC2	2	320x200	4*
		MCGAC3	3	320x200	4*
		MCGAMed	4	640x200	2
		MCGAHi	5	640x480	2
EGA	3	EGALo	0	640x200	16
		EGAHi	1	640x350	16
EGA64	4	EGA64Lo	0	640x200	16
		EGA64Hi	1	640x350	4
EGAMono	5	EGAMonoHi	3	640x350	2
IBM8514	6	IBM8514Lo	0	640x480	256
		IBM8514Hi	1	1024x768	256
HercMono	7	HercMonoHi	0	720x348	2
ATT400	8	ATT400C0	0	320x200	4*
		ATT400C1	1	320x200	4*
		ATT400C2	2	320x200	4*
		ATT400C3	3	320x200	4*
		ATT400Med	4	640x200	2
		ATT400Hi	5	640x400	2
VGA	9	VGALo	0	640x200	16
		VGAMed	1	640x350	16
		VGAHi	2	640x480	16
PC3270	10	PC3270Hi	0	720x350	2

* isti modovi, samo sa drugom "default" paletom boja.

Vrednost varijabli *Gd* i *Gm* za grafičke adaptere i grafičke modove

Listing 1

```
Program DIJAGRAM;
{       Primer upotrebe:
        - DetectGraph;
        - InitGraph
        - GetAspectRatio;
        - CloseGraph,
        Miroslav Olenjin (c) septembar 1991
}
uses  Crt, Graph;

type  NizBoja = array [0 .9] of word;

const Interval = 7;

var   Palete    : dlNizBoja;
      Gd, Gm  : integer;
      Vis, Sir  : integer;
      x, y, z   : real,
      k1, k2    : real;
      Odnos     : real;


procedure InicirajGrafiku;
const Mono  : NizBoja = (0,1,0,1,0,1,0,1,0,1);
      Kolor : NizBoja = (1,9,3,0,5,13,4,12,14,15),
var   Gd, Gm    : integer;
      Xasp, Yasp : word;
begin
      DetectGraph (Gd, Gm);
      If Gd=CurrMono then Palete:=Mono else Palete:=Kolor;
      InitGraph (Gd, Gm, ' ');
      Sir:=GetMaxX div 2;
      Vis:=GetMaxY div 2;
      GetAspectRatio (Xasp, Yasp);
      Odnos:=Xasp/Yasp;
end;


function  f (x,y : real) : real,
begin
      f:=Sin(x)*Cos(y);
end;


begin
      InicirajGrafiku;
      RectAngle (0,0,GetMaxX,GetMaxY);
      k1:=Sir/(Interval*1)/2;
      k2:=sqrt(3)/2
      for x:=0 to 1 do begin
        x:=-Interval;
        while x < = Interval do begin
          y:=-Interval;
          while y < = Interval do begin
            z:=f(x,y);
            Sx:=Round(k1*(x+y));
            Sy:=Round|Odnos*|k1/2*(x-y)+(k1*k2*z)|);
            PutPixel (Sir+Sx, Vis-Sy, Palete[round((z+1)*8 5)]),
            If z=0 then y:=y+k1/0 0 else y:=y+k1/400;
          end;
          If x=0 then x:=x+k1/400 else x:=x+k1/20,
        end;
      end;
      repeat until KeyPressed;
      CloseGraph;
end.
```

Program DIJAGRAM crta tok funkcije *f(x,y)* u tri dimenzije koristeći kolorit da označi lačke na istoj visini

je reda pa se zatim spuštajući za jednu poziciju niže – sve do donje ivice ekrana. Jedno osvežavanje ekrana traje vrlo kratko: oko 1/50 dela sekunde, a kod boljih monitora i manje. Kako se za prelazak mlaza sa kraja jednog reda na početak drugog troši izvesno vreme, to ispada da bi se za razliku od horizontalne, znatnijim povećavanjem vertikalne rezolucije dobila nestabilna treperava slika.

Šta je tu je, mi kao PC programeri moramo svoja remek-dela da prilagođavamo najrazličitijim standardima. Srećom po nas, Borland nam je ne samo u *Turbo Pascal*-u, stavio na raspolaganje nekoliko naredbi za dobijanje parametara vezanih uz određeno grafičko okruženje. Jedna od njih je i procedure

GetAspectRatio (Xasp, Yasp);

One nam u datim dvema promenljivim tipa **word** vraća brojeve koji nas sami po sebi ne zanimaju – za razliku od njihovog količnika! On predstavlja korektivni koeficijent koji moramo da uzmemo u obzir pri iscrtavanju.

Idealno je da ovaj koeficijent bude jednak jedinici, kao u slučaju najvišeg VGA režima kada je rezolucija 640×480. Kod Herculesa on iznosi 0.75, kod EGA 0.775, a kod CGA 0.41. No pazite dobro: ovi brojevi izmereni na jednom računaru ne drugime se ne moraju poklapati jer zavise od toga da li je u njih stvarno ugrađen određeni grafički adapter ili se njegov režim samo emulira od strane neke jeftine kartice! Zato otpada mogućnost da se ove vrednosti unesprede deklarišu i tako preskoči **GetAspectRatio**.

Korektivni koeficijent možete pri iscrtavanju uključiti u proračun na dva načina: ili da ga množite sa **y** koordinatama ili da s njim delite **x** koordinate. Koji način je bolji? Iako ne mora uvek da važi iskustvo je pokazalo da je pogodniji prvi, jer na taj način crtež na može da izađe van granica ekrana kod modova sa nepovoljim proporcijama.

BOJE

Iz date tabele se lakođe može videti i da je broj boja koji pružaju pojedini grafički adapteri veoma različi. Kreće se od minimalno dve kada jedan bit definiše jedan piksel, pa do čak 256 boja, gde piksel određuje ceo bajt. Postavlja se, znači, pitanje, kako sa jedne strane iskoristiti potencijale najjačih kartica, a sa druge obezbediti kompatibilnost sa onim najskromnijim, poput CGA ili Herculesa. I to sa samo jednim univerzalnim kodom, bez pisanja više istih programa specijalizovanih za pojedine kartice.

Ovaj problem je prisutan u većini jezika, ali je barem u *Turbo Pascal*-u lako rešiv, čak i na više načina. Jedan od njih je da u deklarativnom delu listinga definišete tip **NizBoja** (naravno, ime nije bitno) u stvari obični niz od reči, pomoću njega kreirate tako konstante nekoliko skupova boja koje idu uz određene grafičke modove, te još jedan varijabilni radni niz, nazovimo ga **Paleta**. U njega u zavisnosti od vrednosti **GraphDevice** koju je vratila **DetectGraph** procedura presipate najbolje uz pomoć **case** strukture odgovarajuću paletu. Kasnije samo u daljem toku programa ne stavljate **SetColor(Boja)**, već **SetColor(Paleta[Boja])**, i nije neka mudrost.

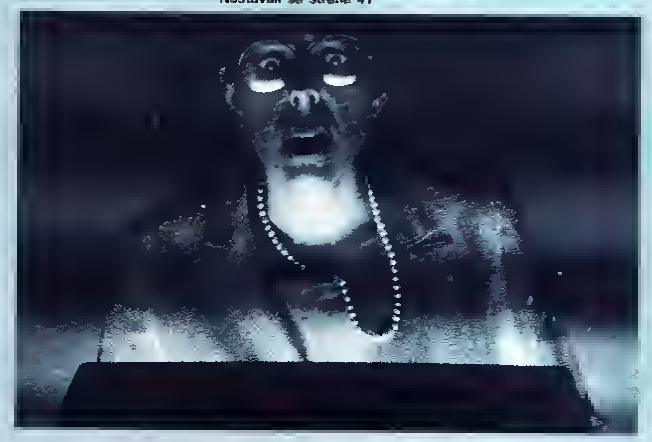
PROGRAM

Priloženi program DIJAGRAM crta tok funkcije **f(x,y)** u tri dimenzije, koristeći kolorit da označi tačke na istoj visini. Dobija se zanimljiva slika, koja podseća na topografske karte.

Manje je, međutim, bitno „šta" radi, dok „kto" radi u svim razlimima svih grafičkih adaptera, upravo zahvaljujući gore opisanim tehnikama. Čitaoci koji imaju SuperVGA karticu moći će da se u to u potpunosti uvere kada program prošeteju kroz sve modove.

SOFTVERSKI FONTOVI NA HERKULESU

Nastavak sa strane 41



font koji se nalazi u nekoj datoteci na disku, da se podesi brzina treperenja kursora i slično.

Kako, dakle, proveriti da li je naš program već instaliran? Jedan od načina da se pronađe segment na koji je učitan naš program prilikom instalacije jeste da pretražimo čitavu memoriju i nađemo karakterističan string, odnosno niz bajtova. Najbolje je da to bude kopirajt koji će se ispisati na ekranu prilikom instalacije. Ovde treba postupiti oprezno zbog jedne stvari. Naime, naš je program dospeo u memoriju sa diska. Pošto se program ne učitava odmah na adresu na kojoj je već sva komunikacija sa diskom ide preko bafera u memoriji se string koji tražimo može pojaviti, dakle, na više mesta. Jednom u baferu za disk, drugi put tamo gde je program prvi put učitan i treći put na mestu gde se program poslednji put učitao i odakle se izvršava. Da bismo izbegli pronalaženje stringa u baferu za disk, kopirajt ćemo na disk zapisati u malo modifikovanoj formi pa nakon startovanja programa vratiti ga u originalni oblik. Za to je dovoljno da prvi karakter bude xor-ovan sa FFh i da se po startovanju obavi još jedno xorovanje (prvo će uraditi sam asembler). Sada string može biti pronađen samo na dva mesta, što je dovoljno da utvrdimo da li je program već instaliran ili ne.

JOŠ NEKOLIKO SITNICA

Ima još par stvari o kojima treba voditi računa, jer čitav proces treba da se odvija u pozadini. Kako osvežavanje celog ekrana može potrajati duže od osamnaestog dela sekunde, interaptu 8 treba omogućiti da se izvršava i u okviru naše rutine a da je ne ometa i ne dozvoliti da ponovo u okviru njega iznova poziva naš potprogram. Tako bi pozivi otišli prilično duboko što bi moglo da rezultira prepunjenjem steka. Zato ćemo postaviti jedan fleg na početku naše rutine i resetovati ga na kraju zvace se **busy**. Drugi fleg, koji ćemo postaviti na **true** ako nakon izvršenja originalnog prekida 8 primetimo da je naša rutina već aktivna (**signal busy**), signalizirače nam da smo sa predugo zadržali na osvežavanju ekrana i da treba dati šansu i glavnoj aplikaciji da obavlja svoj posao. Naravno, pri sledećem pozivu prekida 8 nastavićemo posao tamo gde smo stali.

Omogućićemo još nekoliko parametara za podešavanje rada naše rutine. Pokazalo se potrebnim da i pored pomenutog flega (u listingu nazvanog „begaj") omogućimo korisniku da definiše i koliko se maksimalno sme ispisati karaktera pre nego se prepusti programu da nastavi se radom. Ovime se u stvari određuje koliko će vremena naša rutina oduzimati vremena glavnoj aplikaciji.

Još jedan parametar koji nam može ulepšati rad sa YUHERC-om omogućava nam da isključimo bold prikaz umesto osvetljenog. Nedovoljno je, naime, osam piksela da se čitko prikažu svi znaci iz karaktera seta (malo „m" na primer). Poneki bold može biti od koristi ako ga nama previše na ekranu i treba razlikovati te delove teksta od ostalog (markirani blok u tekst editoru recimo) ali može biti neprijatan ako je čitav tekst bez očiglednog razloga, osvetljen. Opcija **-a0** u komandnoj liniji će postaviti fleg koji neće dozvoliti da se vrši zadebljavanje karaktera.

Priloženi listing pisan je za A86 asembler koji zato što odmah pravi COM verziju programa ne zahteva deklaracije segmenata i slično. Neće vam biti teško da, ako ne koristite ovaj asembler, listing prilagodite MASM-u. Druge njegove specifičnosti su više nego očigledne: **push ax,bx** prevodi se u **push ax** i **push bx**. A naredba **if** implementira se ovako:

```
if z mov ax,100
postaje
        jna labai
        mov ax,100
labaf:
```

Definisana su i dva jednostavna makroa (**loadseg** i **clear**) radi veće preglednosti i lakšeg razumevanja listinga. Integralni listing možete uzeti sa "Sezama".

Program YUHERC možete preuzeti i sa Sezama. Nalazi se u paketu *Hercules Yu Utilities*, zajedno sa još par korisnih programa koji će vam olakšati rad sa YU tekstom na Hercules karbci. Tu je font editor, datoteka sa srpskom ćirilicom za konšćenje sa YUHERC-om program za brzo prevođenje teksta u razne kodne rasporede i grafički lister datoteka. Iz komandnog režima na Sezamu otkucajte DOWN \IBMPC\YU\HYUTIL.ZIP

386 [OS MSDOS, PC] Nova verzije operativnog sistema je divna prilika da se traže nedokumentovane naredbe VER /R, na primer, osim verzije MS DOS a deje i oznaku interne revizije kao i podatak o tome da li je DOS u gornjem bloku memorije (UMB) ili nije – podaci nemaju mnogo veze jedan sa drugim ali je vrlo verovatno neko u *Microsoft* u koristio ovu kombinaciju tokom testiranja DOS-a.

Komanda FORMAT /AUTOTEST je potencijelno još korisnija – formatiraće disketu ne pitajući za njeno ime i ne nudeći formatiranje sledeće diskete [Preuzeto sa Sezama, autor **Kiro Velkovski** (*kvelkovski*)]

387 [OS MSDOS 386, PC] Komanda FORMAT ima još nekoliko nedokumentovanih parametara FORMAT /SELECT radi MIRROR na disku specificiranom u naredbi dok FORMAT /BACKUP rekli bi se, ne radi ništa posebno. Možete li da otkrijete nešto više od toga? [Preuzeto sa Sezama, autor **Zoren Kehler** (*zkehler*)]

388 [OS MSDOS; PC] Prethodni trikovi su nas navelt da malo pogledamo i COMMAND COM Jedina nedokumentovana naredba koju smo pronašli je TRUENAME – izgleda da daje ime tekućeg direktorijuma „zasobilazeći" eventualni JOIN ili ASSIGN

389 [OS WINDOWS; PC] Sa DOS a prelazimo na *Windows 3.0* – minimizirajte sve aplikacije (pa i *Program Menager*) tako da se vidi pozadina držite pritisnut F3 i kucajte WIN3 (mala ili velika slova, svejedno) Zatim otpustite F3 i pritisnite *Backspace* Ekran će se ispuniti e-mail adresama ljudi koji su radili na ovom operativnom sistemu i to kako programera tako i onih koji su pisali dokumentaciju i prateću podršku testirali paket itd. Verovatno ćete prepoznati imena *bilg (Bill Gates)* i *steveb* (*Steve Ballmer* podpredsednik) [Preuzeto sa Sezama, autor **Zoran Kehler** (*zkehler*)]

390 [PJ TPASCAL, PC] Ako pišete složenji program koji, pored ostalog sa READKEY skanira tastaturu Ctrl Break može da vam pokvari posao – umesto da vaš program detektuje tu akciju njegovo izvršavanje prestaje, datoteke su verovatno nelspravno zatvorene i tome slično Treba dakle, onemogućiti Ctrl Break što se može uraditi na više načina

Ako sa držite DOS-a (ili radite na nekom drugom jeziku) pogledajte sliku 1 – vektor 1Bh je usmeren na „praznu" proceduru i tako onemogućen No ako već koristite Turbo Pascal

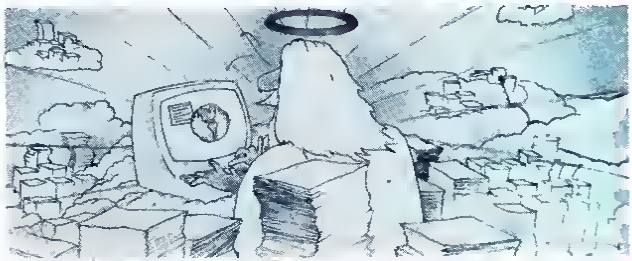
Listing 1

```
procedure dummy;
begin
end;

procedure disablebreak;
begin
  GetIntVec($1b,orgintlb);
  SetIntVec($1b,Addr(dummy));
end;

procedure enablebreak;
begin
  SetIntVec($1b,orgintlb);
end;
```

Vektor Ctrl Break (1BH) u DOS-u pokazuje na praznu proceduru



mnogo ćete „jeftinlje" proći ako konstite njegove mehanizme – jednostavno na početku programa napišite **CheckBreak** =false [Preuzeto sa Sezama autori **Ivan Stanisavljević** (*ivan s*) i **Dragan Zakić** (*dzakic*)]

391 [COMM] Kako da pošaljete poruku nekome ko ima račun na *Compuserve*-u preko BITNET-a (npr sa UBBG-a)? Jednostavno ne njegov *Compuserve* broj dodajte @compuserve com Ukoliko je ne primer, broj 12345 678 puna adresa bi bile

UBBG IN%"12345 678@COMPUSERVE COM"

[Preuzeto sa Sezama autor **Dejan Vučinić** (*vuls*)]

razvoj programa koji će raditi u *Windows* okruženju kao i za konvertovanje DOS programa u programe koji će raditi pod *Windows*-ima

392 [OS MSDOS, PC] U našoj kompjuterskoj štampi je unekoliko zapostavljena „umetnost" pisanja *batch* programa na PC-ju – pravo je čudo šta se sve u tom siromašnom jeziku može uraditi Program sa slike 2, na primer, nazvite C, a onda kucajte C drive dir1 dir2 i naći ćete se u direktorijumu drive \ dir1 \ dir2 baš kao da ste konstili CD i otkucali uobičajenu dvotačku i obmute kose crte – jako zgodno ako vam je YU tastatura „pokrila" znak \ Zanimljivo je da, ako navedete tri direktorijuma od kojih drugi ne postoji, a treći je poddirektorijum prvog program vas automatski prebaci u njega

Ako imate više od 3 particije (C D E) obrišite odgovarajuće REM ove [Preuzeto iz *PC Magazine*-a June 25 1991 autor **Winston Lee, Montreal, Canadaj**

393 [BASE FOXPRO, PC] Pre par meseci pisali smo o problemime koje programen u *Clipper* u moraju da reše kako bi recimo pretvorili niz malih slova u velika uz korektan tretman YU znakove Ako pređete na *FoxBase* ili *FoxPro*, problemi su rešeni praktično sami od sebe i to sve zahvaljujući funkciji SYS(15)

Napišite na primer, izlaz = SYS(15 uzorak, ulaz) pri čemu su uzorci definisani na slici 3 Ovratite pažnju na blankove njihov broj je veoma bitan Za dalju primenu funkcije SYS(15) konsultujte dokumentaciju programskog pakata *FoxPro* [Autor **Branko Meštrić** *mestar*)]

394 [MREZ,NOVELL 342, PC] Pre par meseci pisali smo o automatizovanju procesa prijavljivanja i odjavljivanja sa mreže i problemima koji se javljaju zbog poništavanja *path*-a mapiranja diskova i tome slično Sada dajemo procedure koje će čitav proces učiniti jednostavnim i sigurnim

Instelacija podrazumeva kopiranje datoteka LOGON BAT i LOGOFF BAT sa slike 4 u osnovni direktorijum drajva C I kopiranje datoteke OFFNET BAT u direktorijum SYS \ LOGIN na serveru lokalne *Novell* mreže Svi korisnici koji rade na mreži moraju imeti barem ORS (*Open/Read/Search*) na direktorijum SYS \ LOGIN što se i podrazumeva za grupu EVERYONE po instalacji operativnog sistema *NetWare* Obzirom da svi programi koriste sistemske promenjive koje se upisuju u *environment* treba obezbediti prostor za njih tako što će se u CONFIG SYS ugraditi nešto poput SHELL=C \ DOS \ COMMAND COM /E 512 / p

Upotreba se svodi na ektuviranje detoteke LOGON BAT komendom LOGON IME Ime korisnika je opciono – ukoliko se ne navede, LOGIN EXE će ga naknadno zahtevati Ukoliko je to prvo logiranje na server, mrežni drajveri IPX i NET3 će biti učitani u memoriju nekon čega se poziva program LOGIN EXE Ukoliko je ime korisnika nepoznato ili je loznika nekorektna proces prijavljivanja se prekide Odjavljivenje sa mreže svodi se na pozivanje datoteke LOGOFF BAT odnosno na kucanje komande LOGOFF Posle odjavljivanja direktorijum SYS \ LOGIN ostaje mapiran na prvi nepostojeći drajv na primer F čime je omogućeno ponovno prijavljivanje na mrežu Ukoliko se ne prijavljujemo prvi put, LOGON BAT detektuje postojanje mrežnih drajvera u memoriji te poziva samo program LOGIN EXE u cilju prijavljivanja korisnika

Napominjemo da pre startovanje nekog od programa treba napustiti *Norton Commander* [Preuzeto sa Sezama autor **Zoren Todorović** (*zcodor*)]

395 [IGRE, PC] Više života u sve popularnijoj PC igri *Striker* u STRIKER EXE pronađite niz bajtova C6 06 76 1F 05 a onda umesto 05 upišite onoliko života koliko vam je potrebno na primer FF [Preuzeto sa Sezama autor **Dragan Zakić** (*dzakic*)

Listing 2

```
@echo off
for %%b in (a b c d e f) do if (%1) == (%%b) %%b:
rem for %%b in (0 1 2 3 4 5 6 7 8 9) do if (%1) == (%%b) shift
for %%b in (a b c d e f) do if (%1) == (%%b) shift
rem for %%b in (0 1 2 3 4 5 6 7 8 9) do if (%1) == (%%b) shift
for %%f in (\ %1 %2 %3 %4 %5 %6 %7) do cd %%f
```

***Batch* program za kretanje kroz direktorijume**

Listing 3

```
za TPUNK:   U = STR( 15  UPI  , nis )
za LOGON:   L = STR( 15, LOG , nis )
za FOOOOBT: B = STR( 15, FOO  nis )

UPI = space(47) + "0123456789          " +
                + "abcdefghijklmnopqrstuvwxyzöüä" +
                + "ABCDEFGHIJKLMNOPQRSTUVWXYZÖÜÄ"

LOG = space(47) + "0123456789          " +
                + "abcdefghijklmnopqrstuvwxyzöüä" +
                + "ABCDEFGHIJKLMNOPQRSTUVWXYZÖÜÄ"

FOO = space(47) + "0123456789          " +
                + "abcdefghijklmnopqrstuvwxyzöüä" +
                + "ABCDEFGHIJKLMNOPQRSTUVWXYZÖÜÄ"
```

Uzorci za funkciju SYS(15) u *Fox Pro*-u

Listing 4a

```
rem ;-------------------------------------------------------------------------
rem : Datoteka    :  LOGON.BAT                                                 :
rem : Funkcija    :  Definisi direktorijum drajva C:                           :
rem : Autor       :  Logon <<ime_korisnika>>                                   :
rem : Komentar    :  Datoteka obavlja logiranje korisnika na server NOVELL     :
rem :             :  mreza.  U kombinaciji sa ostalim datotekama omogucava     :
rem :             :  visestruki i konsistentni login i logout raznih baz       :
rem :             :  potrebnih za testiranje softvera                          :
rem : Autor       :  BOBAN TODOROVIC                                           :
rem ;-------------------------------------------------------------------------
@echo off
cls
if %%LOGINAME%%==NNA goto PRAZAN
if %%SERVERNAME%%==NOW goto LOGIN
rem ;-------------------------------------------------------------------------
rem : Ucitavanje LPX i NET3 (NET4) drajvera                                   :
rem : Nadjeni su u direktorijumu koji nije u putanji (PATH) sime se eliminise:
rem : nepotrebno ucitavanje razlicitih verzija LPX-a bi srusilo sistem!      :
rem : Ovde je predvideno ucitavanje u RAM pomocu Quaterdeck-ovog QRAM         :
rem ;-------------------------------------------------------------------------
cd s:\sys\software
s:\sys\qram\loadhi :ge
s:\sys\qram\loadhi set3
cd s:\
rem ;-------------------------------------------------------------------------
rem : Proba~ivanje se prvi mrezni drajv [F: mapirse se STE\LOGIN] i izvrsava  :
rem : nje login exe programa se logiraje sa servera                           :
rem : Obratite lokalni disk ima sina particija, potrebno je promeniti drajv F: :
rem : sa odgovarajuci [npr  d:]                                               :
rem ;-------------------------------------------------------------------------
:LOGIN
f:
login %1
rem ;-------------------------------------------------------------------------
rem : Korisnicki je i uklonjen sa korisnik ili lozinka neprovereni            :
rem ;-------------------------------------------------------------------------
if errorlevel 1 goto GRESKA
rem ;-------------------------------------------------------------------------
rem : Uklonjen je sve sa  skoli sem da ovo treba (korisnik je napravio logiranj :
rem : Proces logiranja se nastavlja sa niskanim i korisnickim login skriptom:
rem ;-------------------------------------------------------------------------
goto END
:ABORT
echo -
echo -  Desetkovan je problem pri ocitavanju drajvera NET3
echo -  Pogledajte kablovske veze  terminatore i T konektore
echo -
goto END
:PRAZAN
echo -
echo -  Korisnik %LOGIN_NAME% je vec logiran na server prazan
echo -
goto END
:GRESKA
echo -
echo -
echo -  Nepoznat korisnik ili lozinka, logiranje nije obavljeno
echo -
@:
goto END
:END
```

Procedura LOGON.BAT za prijevljivenje ne mrežu

396 [OS 4DOS; PC] Već smo pominjali činjenicu da PUSH i POPD *4dos*-a ne rade ako ih pozovemo iz *Norton Commander*-a. Evo i zašto

Pri pozivu bilo koje komande, NC (*Norton Commander*) otvara nov *shell* (startuju ći ono na šta pokazuje promenljiva COMSPEC dakle u našem slučaju *4dos*) koji postavi nove promenljive alocira nov radni prostor i podesi SETDOS parametre (ako su dati u 4START.BTM) 4START se da podsetimo, izvršava pri svakom pozivu 4DOS.COM što znači da se poziva i pre samog AUTOEXEC-a. Neki parametri se „prenose" u svaki naredni *shell* (npr *e*/*as*), dok se drugi ne prenose (npr SETDOS). Tako, ako neko želi da u svakom novom *shell*-u ima isti *environment*, moraće da prelisti uputstvo i vidi šta se od parametara prenosi a šta ne.

Razlog za nepravilan rad PUSHD i POPD je sada očigledan. Pozivom naredbe PUSHD iz NC-a NC poziva *4dos* koji izvrši datu komandu i pri izlasku iz shell-a naravno obriše sve ono što je zapamtio. Ukratko od PUSH-POP pod NC-om nema ništa. Jedino je ostala zanimljive datoteka 4START u koju se mogu smestiti sve one komande koje se odnose na *4dos* ali samo one koje se ne prenose u naredne *shell*-ove [Preuzeto sa Sezama, autor **Miljan Jovanović** (*mjove*)]

397 [STAM LJET3] Kako natereti *HP LaserJet III* da radi kao ploter? Potrebne je jedino odgovarajuća komandne sekvence koja može da glasi: **Esc E Esc &l2O Esc %0BIN;PW 0,1;iW;;PU,SP 0,** Ze povratak u normalan re! zim treba koristi **SP 0,AF, Esc %0A´L**. Ovde je sa **Esc** obeležen uobičajeni *char*(*27*) dok je ´**L** oznaka za kod *form feed* (char(0CH))

Ukoliko se sekvence često koriste ima ih smisla upisati u datoteke START i END a onda koristiti COPY /B START LPT1 odnosno COPY /B END LPT1 [Preuzeto sa Sezama, autor **Bojen Tepavčević** (*bojt*)]

398 [OS.MSDOS;PC]
Ako je prvi znak imena neke datoteke grčko slovo sigma radi se o obrisanoj datoteci -- Io svi znaju. Postoji međutim jedan manje poznat trik -- ako se na kraj imena izvršne daloleke doda znak sa kodom 255 onda se ta daloleka ne može pozvati iz komandne linije premda je vidljiva iz PC Tools-a i samog ispisa direktorijuma. Probajte na primer RENAME FORMAT.COM FORMATX.COM gde je sa X označen kod 255 (Alt 2 5 5). Posle ovoga naredba FORMAT ne radi iako sam program nije izbrisan! [Preuzeto sa Sezama, autor **Andrije Dulić** (*adulic*)]

399 [COMM, VAX] Mnogi korisnici elektronske pošte na nekom od VAX-ova odgovaraju na većinu primljenih poruka odmah tj. *on line*. S vremena na vreme se međutim desi da ih MAIL posle nekoliko otkucanih redova jednostavno „izbaci" iz moda za unos, pa do tog momenta otkucani leksi propadne.

Listing 4c

```
rem ;-------------------------------------------------------------------------
rem : Datoteka    :  SYSTEM LOGIN SCRIPT                                      :
rem : Funkcija    :  Servis Novell mreze                                      :
rem : Autor       :  Automatski (u toku procesa logiranja)                    :
rem : Komentar    :  Ovo je samo primer sistemskog login skripta u Novellu    :
rem :             :  sli se datoteka LOGON.BAT i LOGOFF.BAT.                  :
rem : Autor       :  BOBAN TODOROVIC                                          :
rem ;-------------------------------------------------------------------------
map display off
map errors off
write "Good %GREETING_TIME  %LOGIN_NAME (\n\n"
write "KONFIGURACIJA VASEG SISTEMA"
write "-------------------------------------------------------------------"
write "Vasa ime                    : %FULL_NAME"
write "Fizicki broj terminala u mrezi  : %STATION"
write "Vas operativni sistem       : %OS %OS_VERSION\n"
write "Vreme logiranja             : %HOUR:%MINUTE %AM_PM"
write "Datum logiranja             : %DAY_OF_WEEK  %MONTH_NAME  %DAY, %YEAR\n"
rem ;-------------------------------------------------------------------------
rem : Ovde treba postaviti sve potrebne mapiranje direktorijuma sa drajve     :
rem ;-------------------------------------------------------------------------
map lan s1:=sys:public
map lan s2:=s1:%MACHINE%/%OS/%OS_VERSION
map n:=sys:home/%LOGIN_NAME
if "%LOGIN_NAME" == "supervisor" then map i:=sys:
rem ;-------------------------------------------------------------------------
rem : Ovde treba postaviti eventualno neke potrebnih "environment" varijabli :
rem ;-------------------------------------------------------------------------
dos set LOGIN_INI="%LOGIN_NAME"
dos set STATION="%P_STATION"
dos set LOGTRAN="NW"
rem ;-------------------------------------------------------------------------
rem : Ovde treba postaviti posebne sistemske atatje                           :
rem ;-------------------------------------------------------------------------
dos verify on
drive i:
map display on
rem ;-------------------------------------------------------------------------
rem : odjavna poruka se mreze ako i mreza                                     :
rem ;-------------------------------------------------------------------------
write "\n\nKorisnik %LOGIN_NAME je logiran na Novell server"
```

Pimer ecript datoteke

Listing 4b

```
rem ;-------------------------------------------------------------------------
rem : Datoteka    :  LOGOFF.BAT                                               :
rem : Funkcija    :  Odjavni direktorijum drajva C:                           :
rem : Pozila      :  Logoff                                                   :
rem : Komentar    :  Datoteka testira da li je korisnik logiran  Ukoliko      :
rem :             :  jeste  poziva se OFFNET.BAT i nilje efektivnog odjav-    :
rem :             :  ljivanja sa servera mreze                                :
rem : Autor       :  BOBAN TODOROVIC                                          :
rem ;-------------------------------------------------------------------------

@echo off
cls
if %%LOGINAME%%==NNA goto LOGOUT
Echo -
Echo - Korisnik nije logiran sa server  logout je bezsmislen
Echo -
goto END
:LOGOUT
f:
cd \login
:ffnet
:END


rem ;-------------------------------------------------------------------------
rem : Datoteka    :  OFFNET.BAT                                               :
rem : Funkcija    :  Ste\LOGIN direktorij sa servera                          :
rem : Pozila      :  Pozila se datoteka LOGOFF.BAT                            :
rem : Komentar    :  Datoteka poziva program logout ako i obavlja odjavlji-   :
rem :             :  vanje sa servera mreze i postavlja "environment" vari-   :
rem :             :  jable na nivo, odnosno vrednosti                         :
rem : Autor       :  BOBAN TODOROVIC                                          :
rem ;-------------------------------------------------------------------------
@echo off
cls
rem ;-------------------------------------------------------------------------
rem : Nravy e: je uobicajeno logiran sa PUBLIC direktorijem u Name se nalazi  :
rem : logout exe program                                                      :
rem ;-------------------------------------------------------------------------
cd s:\public
s:logout
set LOGIN=NE
set LOGIN_INI=
set STATION=
cd\
cd\
```

Procedura LOGOFF.BAT za odjevljivenje sa mreže

Tajna je u tome što se daloleka koja se kreira prilikom editovanja poruke sastoji od slogova pri čemu je jedan slog jeden red sve do pritiska ne Enter. Kada dužina reda prede 255 bajta pojavljuje se čuvena poruka *Record too long for user's buffer* i kraj MAIL-a. Dakle ako poruku pišete *on line* sami pritiskajte Enter na kraju svakog reda i sve će biti u redu. Ili koristite REPLY /EDIT pa poruku unesite u svom omiljenom editoru [Preuzelo sa Sezama autor **Radivoje Zonjić** (*zonjic*)]

Pnloge za ovu rubriku šaljite na adresu „Računari" (za „Bajtove lične prirode"). Bulevar vojvode Mišića 17, Beograd ili preko Sezama u vidu lične poruke uredniku (*mail write dejan*).

DEFEKTNE KUGLICE

Kombinacija letnjih meseci, pogrešno štampanog roka (tekst je plaan kao da će „Raćunari 74" izaći 15. jula a oni su, zbog problema sa papirom, izašli 15. avgusta) i, naročito, teške pitalice učinili su da odziv na „problem defektnih kuglica" bude ispod uobičajenog nivoa. Srećom, pitalica nije ostala bez nekoliko kvalitetnih i detaljnih rešenja.

Podsetimo se pre svega, postavke U svaku od šest kutija spakovane su po 24 kuglice, koje bi trebalo da budu teške po jedan gram Bilo koja od tih kutija (možda ni jedna možda i više njih možda čak i sve) međutim sadrži defektne kuglice koje su za miligram teže od standarda Na raspolaganju je veoma precizna opružna vaga koja meri i deliće miligrama Pitali smo koliko je merenja potrebno da bi se locirala kutija sa defektnim kuglicama

Na prvi pogled će, kao i u 67 pitalici biti potrebno samo jedno merenje iz prve kutije se uzme jedna kuglica, iz druge 2 iz treće 4 iz četvrte 8, iz pete 16 i iz šeste 32 kuglice Sva te kuglice se izmere i ako je odstupanje od potrebnih 63 grama na primer 13 miligrama defektne kuglice su u kutijama 1, 3 i 4 ($13_{10}=001101_2$) Navodja je, međutim, u one 32 kuglice koje treba uzeti iz šeste kutije – u njoj (kao i u ostalim kutijama) su svega 24 kuglice! To navodi na zaključak da su za locinanje problematičnih kutija potrebna dva merenja (što je i tvrdilo desetak naših čitalaca) Ipak, na treba se predavati pre vremena

Ukoliko bi se kutija mogla locirati jednim jedinim merenjem, morala bi postojati obostrano jednoznačna veze (bijekcija) između 64 kombinacije kuglica iz kutija i prekoračenja kod merenja svake od tih kombinacija. Preciznije rečeno, moramo da nađemo šest brojeva čijim međusobnim sabiranjima ne mogu da se dobiju dva ista zbira (brojevi 1 2 3 4 5 i 6, na primer ne dolaze u obzir jer iz zbira 7 na možemo zaključiti da li se do njega došlo kao 1+6, 2+5 ili 3+4) Takav zaključak je sasvim dovoljan za pisanje programa sa slike 1 koji će posle par sekundi rada, „proizvesti" jedino rešenje 11, 17 20 22, 23 24 Znači ako iz prve kutije uzmemo 11, iz druge 17 iz treće 20 iz četvrte 22, iz pete 23 a iz šeste 24 kuglice sa sigurnošću možamo da odredimo u kojim se kutijama nalaze defektne kuglice na osnovu samog prekoračenja. Ako je, na primer prekoračenje 48 miligrama defektne kuglice su u prvoj drugoj i trećoj kutiji. Program koji na osnovu prekoračenja određuje lokaciju detektnih kuglica dat je na slici 2

Prvu nagradu od 2000 dinara je zahvaljujući najdelailnijem objašnjenju problema za služila *Biljana Srdanov* iz Zagubice Ostala nagrada (računajući i „modemsku") smo izvukli 1500 dinara dobija *Zlatko Mirazović* iz Mostare 1000 dinara *Zonca Ilić* iz Šapca a nagradu za rešenje dostavljeno preko Sezama *Nikola Miljković* (*miljko*) iz Beograda Pohvale su zaslužili *Predrag Cvetković Đorđe Hohlačov (hohlacov), Slavko Kukrika (skukrika)* i *Zaharije Manojlović*.

TRANSPORTNA PITALICA

Evo jedne, uslovno rečeno, lake pitalice. Šest prijatelja treba što pre da se prevezu od mesta A do mesta B, koristeći pri tome dva motocikla koji „nose" najviše po dva osobe (svako od prijatelja po potrebi ume da vozi) i kreću se brzinom od 60 km/h (prvi motocikl) i 50 km/h (drugi motocikl). Rastojanje između ovih mesta je 90 kilometara, a brzina kojom se pešak kreće iznosi 5 km/h Koliko je vremena potrebno?

Rešenja pitalice zajedno sa programima koja ste napisali, pošaljite na uobičajenu adresu „Raćunari" (za Dejanove pitalice) Bulevar vojvode Mišića 17, Beograd tako da pristignu pre 1 novembra 1991 Sva pisma se korektnim rešenjima konkuršu za novčane nagrade od 2000 1500 i 1000 dinara dok će kupont (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete **poslednje** tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rodjenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj. Osim običnom, rešenje ove pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – konstrie menije REDAKCIJA, PITALICE, REŠENJE ili iz komandnog moda, pošaljite ličnu poruku konsniku pitalice (*mail write pitalice*) Sva rešenja pristigla elektronskom poštom ravnopravno sudeluju u izvlačenju nagrade i konkurišu za **specijalnu nagradu** od 1500 dinara koja će biti dodeljena najboljem rešenju koje je dostavljeno modemom!

Slika 1

```
Program Trazi;
{
   KOLIKO KUGLICA IZ SVAKE KUTIJE?
    prema programu Slavka Kukrike
          "Racunari 76"
}
var w:integer;
    s:array [1..10] of integer;
    x:array [0..10] of integer;

procedure ask(q:integer);
begin
   kom:=0;
   s[q]:=2*(q-1)+1
   repeat
      kom:=kom+1;
      if s[s[q]]=0 then
      begin
         s[s[q]]:=q;
         for w:=1 to 70 do
         if (s[w]>0) and (s[w]<q) then
         begin
            if s[w+x[q]]=0 then s[w+x[q]]:=q;
            if s[abs(w-x[q])]=0 then s[abs(w-s[q])]:=q;
         end;
         if q=6 then writeln('Iz kutija moramo uzeti po ',s[3]-3,
                  s[2]-2,s[2]-1,s[4]-3,s[5]-3,s[6]-3,' kuglica.');
         ask(q+1);
         for w:=1 to 70 do if s[w]=q then s[w]:=0;
      end;
   until s[q]=10+q;
   for w:=q to 10 do s[w]:=0;
end;

begin
   s[0]:=0;
   for w:=1 to 70 do s[w]:=0;
   ask(0)
end.
```

```
   Sa: Potrebno je _____ časova i _____ minuta.

Ime i prezime _____

Adresa _____

Mesto _____

Kompjuter _____ Vreme _____

Identifikacioni broj: |__|__|__|__|__|__|__|__|__|__|
                      za broja tel.  god  rod.  po izboru
```

Slika 2

```
Program Ode_su_defektne;
var a,b,n,q,r:integer;
    g:array [1..6] of 1..24;
    t:array [1..6] of 0..1;

procedure trazi;
begin
   for a:=0 to 63 do
   begin
      n:=a; r:=0; n:=64;
      for b:=1 to 6 do
      begin
         n:=n div 2;
         r[b]:=n mod 2;
         r:=r+t[b]*g[b];
      end;
      if r<>0 then exit;
   end;
   writeln('Takvog prekoracenja ne moze biti');
   halt;
end;

begin
   write('Koliko ste prekoracenja? ');
   readln(r);
   g[1]:=17;g[2]:=17;g[3]:=23;g[4]:=23;g[5]:=23;g[6]:=24;
   trazi;
   if a=0 then writeln('Nema defektnih kuglica.')
          else write('Defektne kuglice se nalaze u 1
   for a:=1 to 6 do if t[a]=1 then write(a,);
end.
```

Program koji na osnovu prekoračenja određuje lokaciju defektnih kuglica

SEZAM 8

Priprema: Dejan Ristanović

BILTEN AVGUST 1991

I pored epitela „letnji" koji obično pripirujemo mesecu avgustu ne Sezamu je bilo veoma živo. I to ne samo u FORUM-u koji je prirodno, prepun poruke u kojima korisnici komentarišu aktuelne događaje – u PC HARD, PC SOFT, KOMUNIKACIJE i ostale stručne konferencije iz meseca u mesec stiže sve više interesantnih i korisnih poruka!

Kada je reč o bilansu, tokom avgusta smo „opslužili" 14 458 poziva, komunicirali sa korisnicima 2 883 časova (!) 120 dana (obzirom da mesec ima 31 dana u proseku je bilo **neprekidno** zauzeto 3 9 nodova) od čega je u *chatu* provedeno 637 časova ili 80 radnih dana (i radne noći). Korisnici su se Sezama *download*-ovali 405 megabajta datoteka dok je za isto vreme asalemu poslato 20 megabajta. Sve ove cifre (osim, začudo, vremena u *chatu*) su daleko veće od julskih.

Što se stvari „koje ostaju" tiče, tokom avgusta je u konferencije pristiglo oko 5300 poruke (6 1 megabajte teksta gotovo duplo više nego u junu) i 3 3 megabajte datoteka priključenih uz njih. 4677 privatnih poruka (oko 1 9 megabajti privatne pošte, ne računajući datoteke uz le poruke). 1606 poruka u grupama (megabajt leksta) i mnogo novih daloteka u direktorijumime, kojima posvećujemo odvojeni Billen.

Na nešem novosadskom nodu koji zahvelujući entuzijazmu *Saše Tepavičkog*, radi eksperimentalno već nekoliko meseci (pre svega radi „ugrevanja" softvera za (buduću) mrezu Sezam nodove po raznim krajevime zemlje) takođe je bilo živo. 510 poziva i 88 časova ne vezi uz *upload* od 856 i *download* od skoro 22 megabajta.

Konferencija **RAČUNARI** se tokom prve polovine avgusta bavita kašnjenjem našeg časopisa – korisnici su bili sžurno obaveštavani o (ne)prisutnju papire i loku šlempe a vesi o izlasku „Računara" 15 evguste je svakako bila jedna od „vesti meseca" na Sezamu. Posle loga poruka u ovoj konferenciji nije bilo mnogo – letnji broj je očigledno dobro primnjen.

"Knvec" ze svojevrsan bum poruke u konferenciji SEZAM je program *SOR, Sezam Off line Reader*, koji je napisao *Zoran Životić*. Program omogućava da, pošlo ste preneli *pnd* sa novim porukama, sve što vas zanima prenesete u posebnu bazu podataka, da poruke čitate praleći lance diskusija, da pišete odgovore i najzad de automatski kreirate *exec* koji će pri sledećem javljanju prosledili Sezamu. Avgust je doneo verziju 2 0 u koju su, na prediog korisnika Sezema, ugrađena brojna poboljšanja. Bilo je i ideja ze sledecu verziju SOR-a, ali više u domenu sitnih izmena i pokušaja da se uštedi prostor na diskovima – poruke se Sezame zbilja ime loliko da malo koji korisnik može sebi pnušti da ih sve čuvel. Diskusije na ovu le mu u denime kada ovo pišemo i dalje traje. SOR se naravno, može slobodno *download*-ovati sa Sezama.

U konferenciji **KOMUNIKACIJE** diskutovalo se o (hardverskom i softverskom) „utišavanju" modema novim 9600 bps ViVa modemima, *script*-ovima za automatsko prijavljivanje na VAX, *Procomm*-u 2 0 koji je prilično razočarenje jer nije ponudio ništa posebno novo čak ni MNP emulaciju o kojoj se pričalo, ograničenjima multitaskinga u *Telix*-u, ubrzanju prenosa podataka pod MNP-om, BBS-ovima u SSSR-u, neizbežnim *Trickle* serverima, zastojima u radu *Bitnet*-a. Za znalno povećan broj poruka u konferenciji **KOMUNIKACIJE** prvenstveno je zaslužna duga i prilično polemički nastrojena diskusija o mogućnostima povezivanja Sezema sa nekom svetskom mrežom (na čemu se ozbiljno radi) i odnosa Sezame i „konkurentskih" domaćih BBS ova.

I u konferenciji **PC HARD** je bilo više poruka nego prelhodnih meseci pre svega zahvaljujući pojavi VGA kartica zasnovanih na ET4000 čipovima lakustva su dobra, ali svaka nova generacije VGA kartice donosi odredene probleme sa instalecijom komercijalnog softvera i programiranjem. Osim o grafičkim karticama, razgovaralo se i o 80486 pločama koje postaju sve jeftinije, entmetičkim koprocesorima, merenju performansi I, naravno, o hard diskovima različitih proizvođača a posebno o *Seagate*-u.

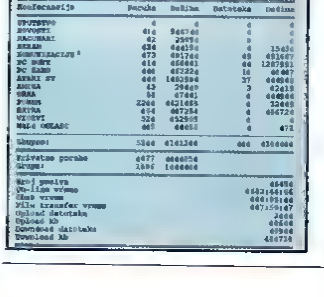
Što se konferencije **PC SOFT** tiče, najvše se pričalo o novom DOS u 5 0 a po malo i o DR DOS-u se istom oznekom. Bilo je mnogo reči i o 4DOS-u toliko da je otvorene i nove lema pod nazivom *4dos*. Dobili smo PD program DFLAT3 (autor je *Al Stevens*) za kreiranje korisničkih interfejsa u programskom jeziku C. Od *Dragana Zakića* (*dzekić*) dobili smo program *YuHerc* koji omogućava upotrebu YU slova na Hercules karticama **bez** hardverske intervencije. U toku celog avgusta mogao se (u lemi *tools*) pratili razvoj ovog programa problemi koje su članovi Sezama imali sa njim i ispravke i dopune autora. Na kraju su problemi „spegljani", a *YuHerc* se pobličio u program izuzelne upotrebne vrednosti. Što se loših vesti tiče, novo tumačenje carinskih vlasti o naplati carine na softver naišlo je na ogorčeni prijem među korisnicime Sezeme.

AtariSTi au se nepokon vratili sa mora, sudeći po broju poruke u lemi *programiranje*, veoma raspoloženi za „hakensanje". Pažnju je zaokupio program koji je uspešno rešio problem instaliranje ekranskog fonta i YU laslatura koji neće zauzimati puno memonje. Pored programerskih lema bilo je dosta govore o PD programu MiNT, koji po prvi pul na ST-u, omogućava previ multitasking. Ovaj program je na rasplegenju korisnicima sezam-a i može se naći u direktorijumu ATARIST\UTILS. Bilo je diskusije i o novom TT emulatoru za *Macintosh*-a. ST ZIP-u o tome da li je *Gemini* bolji od *Neodesk*-a i o nezaobliaznom GDOS-u.

I pored godišnjih odmora, priliv poruka u konferenciju **ORKA** je nešto poraslao – bilo ih je skoro 90. Najviše se govorilo o igrame računarskim zakonicima i HP-ovom računaru 48SX koji je iz meseca u mesec sve zastupljeniji među korisnicima Sezame.

Avgust u **FORUM**-u je protekao apsolutno u znaku rata u Hrvatskoj. Otvorena je nova lema *mir*, u kojoj korisnici daju svoj dopinos za mir i diskuluju o postojećim mirovnim pokretima. Pučˇu SSSR-u je snažno zaokupio pažnju korisnika, s tim da se o njemu promišljalo uglavnom kao o nečemu što je veoma usko povezano sa sudbinom Jugoslavije i Srbije. Što se „nepolitičkih" teme tiče, diskutovalo se o odnosu astronomije i astrologije – diskusija je bile toliko interesantna za naše korisnike da smo je nju otvorili novu temu *zvezde* u konferenciji **EXTRA**.

Trenutna politička situacije u zemlji ni najmanje nije utkala na događenja u konferenciji **VICEVI**. Možda zbog ozog prayila da smo u nevoljama uvek bili najduhovitiji. I dalje stižu najnoviji vicevi a glasanje je i dalje vrlo aktuelno. Interesantna stvar je bila da su neki korisnici uspevali da u ovoj konferenciji diskutuju o stvarnima kojima ju nije mesto. To su redili tako da su uz svaku poruku na neku drugu temu, slali i po jedan vic. Za kraj naravno jedan od viceva: Kako se u Zagrebu kaže frigid na devojka? Ledolina!



KAKO POSTATI ČLAN

Postupak učlanjenja počinje zvanjem Sezama. Nekon poziva broja 648-869 i uspostavljena veza (parametri 2400N81) i pondravne maske, otkloće pitanje:

Username.

Na njega treba odgovoriti se NEW (novi korisnik) nakon čega vas Sezam i niz ineg prosee prijave. Prvi podatak koji treba da date je vaše pun i me i prezime a potom lilrate pseudonim pod kojim ćete koristiti sistem. Pseudonim je bilo kakva reč da 16 znakova koja sadrži samo mala slova i znak (tačka) i ekično ne bira takve da asocira na vaše ime. Sistem će na osnovu vašeg pravog imena predložiti pseudonim koji možete ali i ne morate prihvatiti. Pre izbora pseudonima dobro razmislite, jer on postaje vaše trajno ime na Sezamu i na mnogim ga vše pomoćili.

Sledeći korak je izbor lozinke (password) kojim se vaš pseudonim štiti – prijavljivanje sistemu je moguće samo ako poznajete lozinku koja je vezana uz vaš pseudonim.

Nakon svih navedenih podataka sledi izbor nekoliko radnih parametara (objašnjenje ćete dobiti na ekranu) a potom unos ostalih ličnih podataka – adrese, telefona i podataka o njemi koju koristite za zvanje Sezama. Ovim je prve faze gotova.

Posebno obratite pažnju da **adresu unesete korektno** – ispravna adresa je jedini način da Uprava ostvari daljl kontakt sa vama kako bi se do kraja obavio posao vašeg učlanjenja.

Sledeću radnju dana će vam biti poslato pismo u kome ćete pronaći i popunjenu uplatnicu. Na vama je da kolarecki pošlo na koji se učlanjujete i na osnovu toga na uplatnici unesete iznos predviđen:

za 3 meseca	600 din
za 6 meseci	1200 din
za godinu dana	2400 din

Nakon izvršene uplate (fotokopiju potvrde o uplati poslajte na naznačenu adresu Uprave Sezama bilo poštom ili (daleko bolje) faksom. Po prijemu ove potvrde Uprava će vam iskoj radnog dana otkloriti prelaz sistemu, time postaje punopravan član Sezama, lekoj dan će na vašu adresu biti poslato i Uputstvo za korišćenje sistema.

Ukoliko pri prvom prijavljivanju imate problema i ne završite pojunjavanje podataka korektno nije potrebno da zovete ponovo i prijavljujete se pod novim imenom. Sačekajte jedan dan pa pozovite ponovo – administrator zatekne kod dotašnjeg neispunjenih prijava isbrisaće i daju vam mogućnost da pri sledećem javljanju ponovo popunite prijavu ili vam ostavljaju ličnu poruku kojom vas obaveštavaju šta treba daju da učinite.

Ukoliko ste se ipak prijavili pod više različitih pseudonima, imajte na umu da Uprava sama neće da izvrši isbor koju od vaših prijava će vasti u daju obradu. Kriterijum je najčešće kompletnost podataka. U slučaju više kompletnih prijava uzima se u obzir samo poslednja. U svakom slučaju, na ujlatnici će biti upisan pseudonim na koji se uplata odnosi i samo njega možete koristiti.

Ukoliko želite pre prijavljivanja da razgledate sistem, na pitanje Username odgovorite se DEMO. Nalo ćete se u demonstracionom režimu rada u kome možete pregledati kompletnu strukturu sistema koju je identična onoj pravoj. Broj DEMO poziva nije ograničen a vreme se jirlno sensiju je 5 minuta.

SezamFile

Priprema: Jovan Bulajić

02

Avgust je doneo do sada ubedljivo najveću količinu novih programa za PC – skoro 4 MB. Reorganizacija stabla je donela četiri nova IBMPC podkatalogu – EDITOR, INFO, PROGRAM i TEST. Što je znatno olakšalo snalaženje među *Sezamovim* datotekama. Od prošlog broja dugujemo vam kompletan spisak IBMPC direktorijuma – iako njihova reorganizacija još nije završena spisak dajemo u tabeli 1. Najtraženije programe možete videti na top listi za avgust (tabela 3). Dva od njih ćemo i detaljnije predstaviti u ovom broju, a ostale (nadamo se) u nekom od naredних. Novitete je bilo isuviše da bismo ih sada i nabrajali, a i ne varujemo da bi vam njihovi nazivi nešto mnogo značili pre nego što vam predstavimo bar deo PD softvara koji možete naći na *Sezamu*.

Od ovog broja krećemo sa detaljnijim prikazima programa iz *Sezamovih* direktorijuma. Počećemo sa već mesecima najtraženijim od njih – antivirusnim programom SCAN (VIRUS CAN) kompanije McAfee i njegovim srodnicima – CLEAN, VSHIELD i NETSCAN. Tu je i program domaćeg autora Dragana Zakića (dzakic) koji rešava neke naše specifične probleme vezane za rad sa YU slovima.

Tabela 1

ARCHIVER	Arhiveri
COM	Alat za komunikaciju
DEMO	Demo verzije programa
DRIVERS	Drajveri
EDITOR	Editori tekst procesori DTP
FUN	Zabavni programi igrice itd
GRAPHICS	Grafičke rutine slike itd
INFO	Dokumentacija, prirucnici
MISC	Sve i svasta
PROGRAM	Programiranje, program. alatka
TEST	Dijagnost. testovi, benchmarci
UTILITY	Uslužni programi, razne rutine itd
VIRUS	Zaštita od virusa
YU	Sve u vezi YU slova.

Svi IBMPC direktorijumi

Tabela 2

Grane direktorijuma	Broj datoteka	Ukupna dužina
IBMPC	39	3 808 913
ATARIST	58	1 920 479
Ostalo	10	212 224
Ukupno	**107**	**5,941,516**

Nove datoteke u avgustu

SCAN & Co.

SCAN je bez sumnje najrasprostranjeniji antivirusni program kod nas, a i u svetu. Od velikog nalala računarskih virusa pre dve-tri godine do danas, SCAN se održao na vrhu zahvaljujući svojoj pouzdanosti, redovnim novim verzijama i lakoj dostupnosti (pošlo je *shareware*). Poslednja verzija, koja nosi oznaku 7.6V80, izašla je krajem juna i sposobna je da otkrije čak 193 poznata virusa, a također i njihove podvarijante i klonove (kojih ima ukupno 661 (!)). Mi smo u našem radu naleteli (na svu sreću) samo na nekoliko, a i u samoj dokumentaciji programa se spominje podatak da je za čak 98% zaraza zaslužan neki od svega dvadeset najraširenijih virusa.

Korišćenje SCAN-a je izuzetno lako, po trebno ga je samo startovati sa

SCAN <direktorijum/disk> [/opcije]

i on će prelražiti prvo memoriju (RAM), pa zatim datotake u zadatom direktorijumu ili disku (npr. SCAN C: za skeniranje cele particije). Postoji /ev opcija koja na kraj svake testirane datoteke doda njenu kontrolnu sumu (*validation code*) koje omogućava da se prilikom ponovnog skeniranja (sa opcijama /av ili /cv) odrede eventualne promene u sadržaju datoteka i time otkruje prisustvo nekih novih, za trenutnu verziju SCAN-a nepoznatih virusa. Možete mu zadati opciju za brisanje zaraženih datoteka koje pronađe, ali je daleko bolja rešenje konačenja programa CLEAN.

CLEAN je namenjen popravci datoteka napadnutih (oštećenih) nekim virusom, a samim tim i uklanju virusa sa sistema. Sa CLEAN-om je nešto komplikovanije raditi nego sa SCAN-om – potrebno je prvo skenirati disk i videti ima li zaraženih datoteka, a ukoliko ih ima, SCAN će prijaviti nalaz i ispisati kod (naziv, I.D.) virusa. Zatim je potrebno startovati CLEAN sa opcijama

CLEAN <disk> [kod virusa]

i pustiti ga da radi. Ovo izgleda malo kabasto, ukoliko je vaš racunar napadnut sa više virusa, potrebno je startovati CLEAN posebno za svaki od njih, ali ako vas je već snašla tekva nevolja, nećete se buniti za malo više kucanja. Na primar particija C: i D: diska ćete očistiti od *Vienna* virusa (bio je svojevremeno „popularan" kod nas) sa

CLEAN C: D: [Vienna]

U nekim slucajevima (zavisno od zaraze koje vas je pogodila) neće biti moguća potpuna popravka zaraženih programa, tako da vam preostaje samo da vraćala podatke iz *backupa*.

Ukoliko se baš mnogo plašite virusa (ili mislite samo da ste dosta oprezni) možete instalirati VSHIELD – program za prevenciju zaraze. Instalirace se kao rezidentan i kontrolisati svaki program koji se startuje. Ovo uz neizbežan gubitak memorije (zauzima od 6 do 31 KB zavisno od nivoa zaštita koji odaberete) donosi i usporenje od nekoliko sekundi prilikom učitavanja svakog programa. Ukoliko VSHIELD pronađe virus u nekom od startovanih programa neće mu dozvoliti da se instalira i upozorice vas na zarazu.

NETSCAN je varijanta SCAN-a za prelražvanje po računarskim mrežama. Prelražuje sve mrežne diskove i podržava sve poznate mreže.

HYUTIL

Svi vlasnici Hercules grafičke kartice (a te kvh je još uvek većina) znaju kakve su muke sa YU slovima. Ukoliko su vam potrebna morale ih ugraditi u EPROM da biste ih isključili, morale se propinjati do prekidača sa zadnje strane računara, o promeni kodnog rasporeda (bez preprogramiranja EPROMA) možete samo da sanjate, a ukoliko slučajno nemate ugrađen prekidač osuđeni ste na celodnevno gledanje naših slova tamo gde im mesto nije. Što je pravo mučenje naročito ako imate slova ugradena po YUSCII standardu (a lakvih je opet većina). Dakmićno (a uz ovakvu arhitekturu Herculesa jedino moguće) rešenje ih problema je prelazak u grafički mod rada, ali to većina programa ne radi, a uz to je takav rad i dosta sporiji.

HYUTIL je zbirka programa koji rešavaju neke od vaših problema. YUHERC je glavni modul i namenjen je prikazivanju naših slova na Herculesu. Program se služi alegantnim trikom – instalira se kao rezidentan i prebacuje u grafički mod pušta ostale programe da pišu po fiktivnom tekstualnom ekranu, a sam obezbeduje prikazivanje i osvežavanje grafičkog ekrana, na kome su neravno prikazana i sva naša slova. Možete birati kodni raspored koji vam odgovara (podržano je pet – YUSCII). Računari Cuetom 2, Latin 2 i Moj Mikro) i editovati fonl ukoliko želite neki svoj raspored ili oblik slova (a uz standardan priložen je ciniični fonl). U toku rada prilikom na magične tastere možala prelazati iz grafičkog u standardni (teksl) režim i obratno. Testirali smo rad sa mnogim popularnim programima i većina ja radila kako treba. Imali smo problema sa DesqView om, ali smo zaključili da je on isuviše „prljavo" pisan da bi se tu moglo nešto uraditi). Kao rezidentan, YUHERC zauzima oko 4.5K memorije.

Z je program nalik na čuveni Isl – namenjen je brzom pregledu tekstualnih datoteka. Razlika je u tome što Z radi u grafičkom modu i tekst prikazuje u rezolucji 35 redova * 90 znakova što doprinosi preglednosti a i su i YU slova.

YUCONV služi za konverziju teksta iz jednog u drugi YU kodni raspored, što znači da je namenjen svima a ne samo vlasnicima Herculesa. Konverzija je moguća iz bilo kog u bilo koji raspored i odvija se vrlo brzo.

Autor HYUTIL-a je Dragan Zakić (dzakic na *Sezamu*).

TOP 10 Datoteka – AVGUST 1991

	direktorijum	datoteka	opis	
1	\ COM	sor100a	zip	*Sezam* Offline Reader – Verzija 1.00A
2	\ IBMRC \ YU	hyutil	zip	Hercules Yu Utilites v 1.2 (dzakic)
3	\ IBMPC \ VIRUS	scanv80	zip	McAfee virus detektor
4	\ IBMPC \ FUN	dash	zip	Boulder dash
5	\ COM	sor100u	zip	Update sa SOR 1.00 na SOR 1.00A
6	\ IBMPC \ ARCHIVER	arj220	exe	ARJ arhiver (v 2.2)
7	\ IBMPC \ PROGRAM	dflat3	zip	C biblioteka za korisnicki interfejs (DDJ)
8	\ IBMPC \ YU	foggys	zip	FOGGY SingleFont VGA Resident Builder (gbrodic)
9	\ IBMPC \ FUN	corwp21	zip	CoreWar simulator
10	\ IBMPC \ VIRUS	clean80	zip	Uklanja viruse koje Scan nađe

Tabela 3 Najtraženije datoteke u avgustu

SPECIJALNI KOMERCIJALNO-INFORMATIVNI BLOK

• akcije • marketing • publicitet

U OVOM BLOKU PREDSTAVLJAJU SE:

IBIS-SYS Beograd	2	GOSTOL, Novica Golica	66
PC PROGRAM Beograd	57	MICROSYS, Novi Sad	67/69/71
YUCO Novi Sad	57	MACE Beograd	68
INTERSOFT, Beograd	57	EUROHIT Beograd	68
INFORTRADE, Koper	57	MZ, Beograd	70
GAMA, Beograd	58	MOST Beograd	72/73
MLACOM, Ljubljana	59	MIKRONOVA, Novi Sad	74
ŠUTULIĆ, Beograd	60	OSA Beograd	78
COMTEC Novi Sad	61	JUGOTRADE Novi Sad	79
QUANTUM Ljubljana	62	ENERGODATA, Beograd	80
PERIHARD, Beograd—Zagreb	63	COMTRAD Dusseldorf	81
COMPUTER HIT, Ljiđa	64	PEPRETUUM, Zagrad	82
BORIS KIDRIČ Beograd	65	MCH Maribor	83
JUGOSL KNJIGA Beograd	66	ATLANTIS Ljubljana	84

Za sve komercijalne aranžmane izvolite nam se javiti na:
tel. 011/653-748, fax 011/648-140, modem 011/648-899
(osoba za kontakt Esad Jakupović)

Turbo Pascal - naredbe i objekti
Autor Ouško Savic
Jednostavan uvod u objektno programiranje!
Nasleđivanje virtuelni metodi, klase, pisanje novih objekata, opis svih 300 naredbi, prevod svih grešaka, ilustracija DOS i BIOS poziva, opis direktiva prevodiocu, testiranje programa
Svi primeri napisani su korišćenjem objekata
Priručnik za verzije 5.5 i 6.0 - preporučuje ga i Borlendov zastupnik za Jugoslaviju!
Strana 240, januar 1991, ISBN 86 901 167 1 0
Cena 300 dinara + poštarina. Isporuka pouzećem istog dana!
PC Program, Generala Henrika 20, 11040 Beograd, tel/fax (011) 463 296



MALI OGLASI

Običan mali oglas do 10 reči staje 900 dinara, a svaka dalja reč 80 dinara. Centimetar uokvirenog oglasa staje 1 000 dinara s tim što se mogu zakupiti najmanje 2 cm i što oglas ne može biti viši od 5 cm. Rok za slanje do prvog u mesecu.

KOMPJUTER SERVIS
— SPEKTRUM
— COMODORE 64 i periferija
— AMIGA 500
— PC
Avalska 6 (na Slaviji)
tel 011/434-267

C — VODIČ. Integrisani vodič kroz programski jezik C koji omogućava da se na interaktivan način veoma brzo i lako ovlada ovim programskim jezikom. Vodič, editor, kompajler i help u svakom trenutku su na raspolaganju. Detaljne informacije na tel. (030) 36-583. Isporuka pouzećem. Cena 900,00 dinara. M. Karabašević, NSC N14/8, 19210 — BOR

ASTROSOFT
Programi za izradu horoskopa na XT/AT računarima
Tel. 011/167-606

DISKETE
svih vrsta NASHUA, BASF, računare i dijelove sa garancijom prodajem po najpovoljnijim cjenama. Količinski popusti. ***Tel. (061) 265-525***

DISKETE garancija
TEL. (061) 267-632

5 25" — 2S/DD (360 Kb)	28 din kom
5 25" — 2S/HD (1.2 Mb)	44 din kom
3 5" — 2S/DD (720 Kb)	34 din kom
3 5" — 2S/HD (1 44 Mb)	56 din kom

Na lageru su i BASF i NASHUA

DISKETE IMAJU GARANCIJU ŠTO ZNAČI 100% ERROR FREE

BRZA ISPORUKA, NA VEĆE KOLIČINE POPUST

AMIGA
500, 2000, RAM. Razširitve svih vrsta, disketne enote i ostale dijelove za Amigu prodajem. Tel. (061) 263-813
AMIGA

INTERSOFT
Andrićev venec 2/1
Ročunari
PC AT 286/386/486
Telefax
Panasonik KX-F90
Matrični štampači
EPSON (A4,A3)
TEL.
011/340 408
011/339 104

INFOTRADE
INŽENIRING IN RAZVOJ INFORMACIJSKIH SISTEMOV

INFOTRADE IZOBRAŽEVALNI CENTER u Kopru, Vojkovo nabrežje 30a, organizira sledeće tečajeve za Novellove mikroračunarska mreže za razdobje od oktobra do decembra 1991

NOVELL

TEČAJ	TRAJANJE DAN	OKT	NOV	DEC
1. Predgled karakteristika i mogućnosti NetWare operativnih sistema 286 i 386	1	1	4	2
2. Uvod u mikroračunarske mreže	1	7	5	3.
3. 286 — Upravljač mikroračunarskih mreža	3	2	6	4
4. 386 — Upravljač mikroračunarskih mreža	3	6	12	9
5. Novell — printanje	1	11	18	24
6. Instalacija NetWare 286 — workshop	2	14	18	16
7. Instalacija NetWare 386 — workshop	2	21	20	12.
8. Novell — tehnička podrška — workshop	3	23	25	16

Vaše prijave i sve dodatne informacije o tečajevima možete dobiti na adresi:

INFOTRADE KOPER JAKA PLATIŠE 13 **TELEFON** 064/329-523
PE KRANJ 64000 KRANJ **TELEFAX** 064/323-582

GAMA Electronics Beograd
Mišarska 11
Tel 011/332-275; Fax 011/335 902
Radno vreme pon petak 9-17
GAMA Electronics Trade Handels GmbH
Tel 99/49/89/577-209,
Fax 99/49/89/570-4379



GAMA KOMPJUTERI - STANDARD KOJI PREVAZILAZI VAŠA OČEKIVANJA

Ugled koji GAMA na našem tržištu ima, direktno proističe iz naše neprestane težnje da vam ponudimo kompjuter u koji se bezrezervno možete pouzdati.

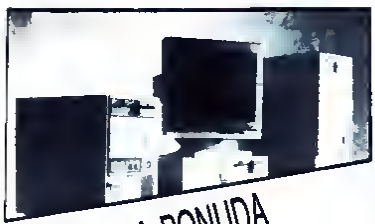
Da smo na dobrom putu svedoče:
garancija od 24 meseca
i
naše dugogodišnje prisustvo i stalan rast na jugoslovenskom i nemačkom tržištu kompjutera.

GAMA KOMPJUTERI BRZI SU I SNAŽNI KOLIKO TO VAMA TREBA, A POUZDANI UVEK I PRE SVEGA

MLAKAR & CO
AUSTRIJA

Računare prodajemo u KIT verziji (u delovima). Za sve uređaje nudimo garanciju, a montaža i servis u Jugoslaviji. Za savete kod izbora pozovite nas na telefon: 9043/4227-2333. Naša trgovina je u Austriji, u Podgori (Unterbergen) kraj glavnog puta prema Celovcu, 60 km od Ljubljane i 12 km od Ljubelja. Trgovina je otvorena od 8 do 17 časova a u subotu od 8 do 13 časova.

FAKS 9943/4227-2091 TELEKS 422749 MLCO A



IZUZETNA PONUDA
RAČUNAR + ŠTAMPAČ

PC-M 286-12-45 S ŠTAMPAČEM
– AT 286/12 MHz, 1 Mb RAM, japanski brzi disk 45 Mb FDD 5,25" 1,2 Mb (ili 3,5" 1,44 Mb) tastatura 101 14" monohromatski monitor štampač Citizen, 9 igl A4

DIN 49.900

– s štampačem formata A3

DIN 60.200

PC-M 286-16-45-NEAT S ŠTAMPAČEM
– AT 286/16 NEAT MHz, 1 Mb RAM japanski brzi disk 45 Mb FDD 5,25" 1,2 Mb (ili 3,5" 1,44 Mb) tastatura 101 14" monohromatski monitor štampač Citizen 9 igl A4

DIN 50.100

– s štampačem formata A3

DIN 61.400

PC-M 386-SX-45 S ŠTAMPAČEM
– AT 386 SX/16 MHz-SX, 1 Mb RAM japanski brzi disk 45 Mb FDD 5,25" 1,2 Mb (ili 3,5" 1,44 Mb) tastatura 101 14" monohromatski monitor štampač Citizen 9 igl A4

DIN 66.800

– s štampačem formata A3

DIN 71.000

Garancija 24 meseci

MLACOM d.o.o
Celovška 185
66100 Ljubljana 1
Tel 061/193-010 Fax 061/193-016
061/553-276

KUĆIŠTA SA NAPAJANJEM	DEM	BODOVI
AT BABY	117	187,20
SLIM	154	246,40
MINI TOWER	167	262,40
TOWER	255	408,10
FILE SERVER 375W	950	1 520,00
WORKSTATION	192	306,80

OSNOVNE PLOČE		
XT 4,77/10 MHz	115	184,30
AT 286-12MHz	145	233,20
HEADLAND 286 16MHz	185	297,20
HEADAK 286-20	235	376,00
386-SX-16 MHz	540	622,44
386-SX-16 MHz ALL-IN-ONE	599	712,60
386-SX-20 MHz	590	792,00
386-25MHz CACHE	1 290	1 368,00
386 33MHz, CACHE	1 560	1 758,40
486/25 MHz	3 300	4 792,00
486-25 MHz 64 KB CACHE	3.500	4 864,00
486-25 MHz, 64KB CACHE, EISA	5 271	6 434,00
486 33 MHz 128 KB CACHE INTEL	3 800	5 104,00

DISPLAY KARTICE		
Printar Hercules	28	44,80
Printer/Hercules/CGA	38	82,60
VGA 800X600/16 bit	124	184,00
Super VGA 1024x768	160	233,00
GENOA 6200 SUPER VGA	255	408,00
GENOA 6300 SUPER VGA	286	425,10
GENOA 6400 SUPER VGA	420	672,00
GENOA 6400 A SUPER VGA	570	912,00
GENOA 6400 V SUPER VGA	495	792,00
GENOA 6200 VC SUPER VGA	660	1 056,00
VGA 1280 × 1024 (NEC 5D)	3 427	5 483,00

KONTROLERI		
HDD XT MFM	60	84
FDD/HDD AT MFM 1:1	70	94,90
DTC 7280 AT MFM 1:1	187	299,40
DTC 7287 AT RLL 1:1	197	315,40
AT(IDE) BUS FDD/HDD	35	56,00
SCSI FDD/HDD	82	131,20
ESDI FDD/HDD	280	448,00

DODATNE KARTICE		
MULTI I/O XT	66	104,90
I/O AT (SER. PORT)	22	32,00
I/O AT (PAR/2×SER PORT)	30	48,00
I/O AT (PAR/2×SER GAME)	33	52,80
MULTI USER (4×RS232)	165	270,40
MULTI USER INTELLIG (8×RS232)	713	1 140,80
AD/DA 12 bits	137	219,40

LAN		
Ethernet compat (NE 1000) B,8bit	235	376,00
Ethernet compat (NE 2000) B 16bit	280	448,00
Ethernet Pocket Adapter	816	985,10
Ethernet boot rom for NE 1000	10	22,60
Ethernet boot rom for NE 2000	10	22,60
IEEE802.3 transce ver	212	340,60
BNC 50 ohm terminator	9	9,10
BNC 93 ohm terminator	6	9,10
N series 50 ohm female terminator	9	13,70
Cable RG 58 (1M)	3	4,60
Cable connector	6	9,10
Ethernet IEEE802.3 repeater	1 207	1 931,40
Arcnet coax-star LAN card	125	201,10
Arcnet coax bus LAN card	138	221,70
Arcnet twisted pair star LAN card	139	221,70
4 port cnpaxial act ve hub card	376	502,00
4 port twisted pair hub card	378	249,10
Remote boot rom for arcnet card	10	16,50
Cable RG 62 (1M)	3	4,80

TASTATURE		
102 tipke	58	92,80
101 tipka click Chicony YU	78	107,20
101 tipka s mrkim Chicony	167	209,60
101 tipke Cherry	138	206,40

MEKI DISKOVI		
5,25 360 Kb	111	177,60
5,25 1,2 Mb	116	184,00
3,5 1,44 Mb	115	184,00

TVRDI DISKOVI		
Seagate – 157A 45Mb/28ms	390	589,90
Seagate – 274A 63Mb/28ms	750	1 299,20
Seagate – 296N 65Mb/25ms	690	1 104,00
Seagate – 1102A 86Mb/19ms	710	1 136,00
Seagate – 1126N 107Mb/15ms	1 408	2.250,30
Seagate – 1144A 125Mb/19ms	990	1 584,00
Seagate – 2182E 160Mb/15ms	2 141	3 426,30
Seagate – 1201A 177Mb/15ms	1 495	2 387,70
Seagate – 2209N 183Mb/18 ms	2 140	3 424,00
Seagate – 1239A 211Mb/15ms	1 600	2.560,00
Seagate – 2363A 338Mb/16ms	2 540	4.224,00
Seagate – 2383E 338Mb/15ms	2.855	4 569,10
ST 2383N 337 MB/14 MS	2 855	4 569,10
ST 2502N 442 MB/16 MS	3 566	5 709,70
ST 4376NV 330 MB/16 MS	3 000	4 800,00
ST 4366NV 337 MB/19 MS	3 405	5 448,00
ST 2502NV 442 MB/16 MS	3 735	5.976,00
ST 4766NV 676 MB/15MS	3 945	6.312,00
ST 41200NV 1.050 MB/15 MS	5 713	9 140,00
NEC		
D 3741 A 44 MB/23 MS	540	784,00
D3661 135 MB/23 MS	1.350	2 160,00
D 5655 175 MB/18 MS	1 490	2 384,00

MONITORI	DEM	BODOVI
14" monohromatski	175	272,00
VGA monohromatski	228	356,20
VGA Color 1024×768	596	860,80
15" AA full size VGA	1 540	1 886,00
NEC Multisync 2A, 3D, 4D, 5D		
VGA CITIZEN 14" 1024 × 768	1.200	1 920,00
PANASONIC 20" 1280 × 1024	4 427	7 083,40
MITSUBISHI VGA 720 × 400	975	1 560,00
MITSUBISHI 14" MULTISCAN	1 350	2.160,00
MITSUBISHI VGA 15" 1024 × 768	3 300	5.280,00
QUME VGA 14" 1024 × 768	1 071	1 714,00

ŠTAMPAČI		
1 CITIZEN 180D, A4	385	484
2 CITI 9 Pin A3	838	1 076,30
Star LC 20	405	648,00
Star LC 15	733	1 104,00
Star LC-24/200	588	1 064,00
Star LC/24 15	1 010	1 616,00
Star ostali modeli		
Laser HP JET III P		4 569,00
Laser HP JET III		6 080
Laser HP JET IIISi		16 240
CANON serija BJ		
QUME serija CRYSTAL PRINT		

ČITAČI		
ROLAND DXY 1100 A3	1 990	2 672,00
ROLAND DXY 1200 A3	2.088	3.341,70
ROLAND ostali modeli		

MODEMI		
2400 int	193	292,80
2400 ext. (MNP5)	249	384,30
9600 ext (MNP5)	1 081	1 577,10
2400 POCKET	227	363,40

UPS – NEPREKIDNO NAPAJANJE		
UPS 300 VA	480	768
UPS 500 VA	570	912
UPS 1000 VA	1 097	1 755

RAM		
41256-10	3	4,80
41256-08	3	4,80
44256-08	11	17,60
411000-08	10	16,00
SIMM/SIP 256K×3/08	35	56,00
SIMM/SIP 1MB×8/08	97	155,00

COPROCESSOR		
80287	279	786
80387SX 16MHZ	650	1 040
80387-25MHZ	790	1648
80387-33MHz	1 080	1.760

STREAMER		
COLORADO 40/60/120 Mb int	795	1.192
TARGA 150 Mb ext.	1 729	2,765

RAZNO		
PC NOTEBOOK XT 20 Mb	1 990	3,184
PC NOTE BOOK 286 VGA, 20 Mb	3 700	5,040
FAX NISSEI	1 148	2.011
FAX CARD	567	981,40
FAX MODEM CARD		
FAX MODEM POCKET		
Čitač bar koda	466	746,10
Prenosni čitač bar koda	1 042	1 667,70
CCD Scanner		
Štampač bar koda		
Miš Genius 6-Plus	66	108,80
Miš Genius GM F-302	97	136,20
Miš bežični	188	289,70
Track Ball	66	104,80
Tablet Genius GT 1212B 12×12	532	851,20
Tablet Genius GT 1612D	1 027	1 589,90
Scanner Handy Genscan GS-4500	290	456,00
Scanner A4 Hanvly w/pap feeder	1 120	1 792,00
Scanner EPSON GT-4000 Color	2.850	4 560,00
Eprom UV Eraser	260	416,00
Eprom Writer Card 4×	392	628,00
Disk Box 5×5,25"	2	3,40
Disk Box 10×5,25"	4	5,70
Disk Box 50×5,25"	12	18,20
Disk Box 5×3,5"	3	4,80
Disk Box 10×3,5"	3	4,80
Copy Holder	14	22,60
Pokrivač za monitor i tastaturu	13	30,20
Sve vrste EPROM		

DEM su cene bez poreza na promet kod Mlakar & CO Austrija

Cene kod Mlacoma su bez poreza na promet – 1 bod vrednost DEM na Ljubljanskoj berzi

Na zalihi i druga oprema.

PC AT 286, 386-SX, 386, 486
ISPORUKA ODMAH

MODEL	SC 2016	SC 2025!!!	SC 3020 SX	SC 3033 CACHE 64 Kb	SC 4033 CACHE 64 Kb
PROCESOR	INTEL 286	INTEL 286	INT 386 SX	INTEL 386	INTEL 486
CHIP SET	U M C	HEADLAND	HEADLAND	U M C	OPTI
RADNI TAKT	16 MHz	25 MHz!!!	20 MHz	33 MHz	33 MHz
LAND TEST	21 MHz	33 MHz!!!	26 MHz	56 MHz	150 MHz
KUĆIŠTE	BABY AT	BABY AT	BABY AT	BABY AT	BABY AT
NAPAJANJE	200 W	200 W	200 W	200 W	220 W
RAM	1 Mb	1 Mb	2 Mb	4 Mb	4Mb
MAX RAM	5 Mb	8 Mb	8 Mb	16 Mb	32 Mb
FLOPPY	1 2 Mb	1 2 Mb	1 2 Mb	1 2 Mb	1 2 Mb
HARD DISK	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb	ST 157 A 44 Mb
HD FD CONT	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1	AT BUS INT 1 1
CORETEST	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S	T/T 1 ms 740 K/S
SER INT	2 RS 232	2 RS 232	2 RS 232	2 RS 232	2 RS 232
PAR INT	2 CENTR	2 CENTR	2 CENTR	2 CENTR	2 CENTR
TASTATURA	101 ALPS	101 ALPS	101 ALPS	101 ALPS	101 ALPS
GRAF KART	HERCULES	HERCULES	HERCULES	HERCULES	HERCULES
MONITOR 14	AMBER ili CRNO-BELI	AMBER ili CRNO-BELI	AMBER ili CRNO-BELI	AMBER ili CRNO-BELI	AMBER III CRNO BELI
CENA DIN.	**45000,00**	**51000,00**	**64000,00**	**90000,00**	**125000,00**

16 " COLOR MONITOR EIZO 9070S (1200*800)		
I VGA CARD 1 MB 1024*768	61000 00 DIN	*
14 " VGA COLOR MONITOR I VGA CARD 512 Kb 1024*768	24000 00 DIN	* (ZVEZDICOM SU OZNAČENE DOPLATE U ODNOSU NA OSNOVNE KONFIGURACIJE)
HD 85 Mb MAXTOR 7080 A, 17 ms	13000 00 DIN	*
HD 130 Mb ST 1144 A IDE 18 ms	18000 00 DIN	*
HD 210 Mb CDC IDE 15 ms	36000 00 DIN	*
1 Mb RAM	3500 00 DIN	
IIT 287—10 MHz KOPROCESOR	6500 00 DIN	(100—300% BRŽI OD INTELA 287)
IIT 287—20 MHz KOPROCESOR	8000 00 DIN	(100—300% BRŽI OD INTELA 287)
IIT 387—33 MHz KOPROCESOR	16000 00 DIN	(100—300% BRŽI OD INTELA 387)
FLOPPY DISK 1 44 Mb TEAC	4500 00 DIN	
MIŠ GM 6000 (RS 232)	2500 00 DIN	
EPSON LX 400 (A4,9 PINA)	pozovite nas!	(PRINTERI SU SA UGRAĐENIM YU ZNAKOVIMA ŠŽĆČĐ I KABLOM ZA VEZU SA RAČUNAROM)
EPSON LQ 550 (A4,24 PINA)	pozovite nas!	
EPSON LQ 1010 (A3 24 PINA)	pozovite nas!	
EPSON LQ 1050 (A3 24 PINA)	pozovite nas!	
EPSON FX 1050 (A3,9 PINA)	pozovite nas!	
EPSON LX 1050 (A3,9 PINA)	pozovite nas!	

ZADRŽAVAMO PRAVO KOREKCIJE CENA BEZ NAJAVE!!!
NAČIN PLAĆANJA: 100% AVANS — ISPORUKA ODMAH.
ŽIRO RAČUN: 60815-601-64787

PREDUZEĆE „ŠUTLIĆ“ & „MIKRO DIZAJN“

KOSOVSKA 32/1. SPRAT, 11000 BEOGRAD,
RADNO VREME: 9—17 h, SUBOTOM 10—14 h

TEL./FAX: 011/332-607

GARANCIJA 12 MESECI, SERVIS I ODRŽAVANJE
BEOGRAD 01. 10. '91.
CENE OPREME VAŽE ZA PRAVNA I FIZIČKA LICA.
(OD 1. 2. '91. PLAĆA SE POREZ 3%)

ComTec d.o.o.

Preduzeće za informatički inženjering i automatizaciju
Novi Sad, Vojvodanskih brigada 7/IX
Telefon/Fax: 021/22.258

LAPTOP - NOTEBOOK



VERIDATA EL 386
samo 2kg!

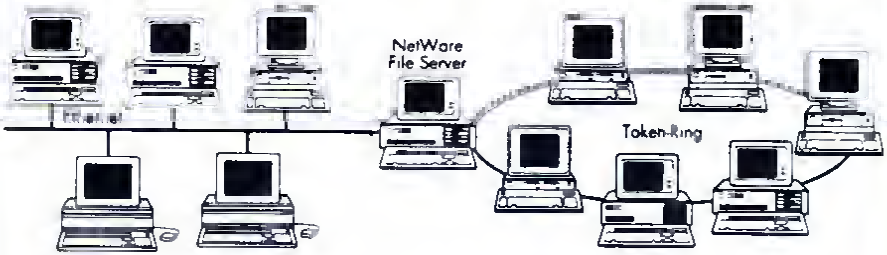


FLYER 286 FH
FLYER 386 FH



VERIDATA LP 286
VERIDATA LP 386

NOVELL



ComTec d.o.o. je autorizovani prodavac *NOVELL* proizvoda
Nudimo Vam i standardne AT 286 /386/486 računare i š irok izbor aplikativnog softvera prilagođenog radu u mreži

Dileri, javite se! Povoljan rabat!

QUANTUM

D O O Stegne 25, 61000 Ljubljana

tel. 061/191-133 int.: 21, 51 — 061/191-740 fax. 061/192-566

Kod nas je svakog meseca nešto novo. Tako smo od 1. septembra 1991. godine postali i distributeri ARC-Net mrežnih proizvoda

firme SMC STANDARD MICROSYSTEMS CORPORATION

vodećeg proizvođača ARC-Net mrežnih kartica.

Još smo, tokođe, provo adresa zo tvrde diskove, kontrolere i mrežne kortice WESTERN DIGITAL i streomerje COLORADO.

WESTERN DIGITAL

tvrdim diskovima veruju: Olivetti, ALR, Siemens/Nixdorf, itd.

Da li vam je svakadnevni backup muka i suviše? Upatrebljavajte streamere

COLORADO MEMORY SYSTEMS INC

QUANTUM

D O O Stegne 25, 61000 Ljubljana

tel. 061/191-133 int.: 21, 51 — 061/191-740 fax. 061/192-566


ScanJet
LaserJet
HEWLETT PACKARD ASORTIMAN
IIIP
IIID
IIISI
III
NOVO!
RECOGNITA
U BOJI
PUNJENJE TONER KAZETA
CRNO
LATINIČNI I ĆIRILIČNI FONTOVI
BIT-MAPIRANI ILI EPSON I SCRIPT
ŠKRAMSKO PRINTERSKI
FONT KAZETE
Kalkulatori
PERI HARD®
YUGOSLAVIA
NOVO!
UPS
STALCI ZA PRINTERE OD PLEKSIGLASA
KUTIJE ZA DISKETE
ZA ARHIVIRANJE
ZA SLANJE POŠTOM
NOVA ADRESA!
PERIHARD ZAGREB
Klekovacka 28/I pp 5030, Zagreb
tel 041/263-696, fax 041/263-714
PERIHARD INŽENJERING
Ivana Milutinovića 24, Beograd
tel 011/436-019, 423-383, fax 011/435-513
STAR LIGHT
OPTIK GLAS
STAKLENI
FORTUNA
UP 'N DOWN
PROFESIONALNI ZAŠTITNI FILTERI
PROTECTOR
SUNFLEX
MIKROVLAKNA
FLEX
VEC
HEWLETT PACKARD
LaserJet IIIp

Computer hit biblioteka
Poštenski fah 116, 71210 Ilidža
Tel. 071/ 270-213 i 071/621-025

Computer hit biblioteka
LITERATURA ZA IBM PC

Novo u izdanju Computer hit biblioteke
Edicija "Džepna knjiga za PC"

1. MS DOS 3.3/4.0

Knjige neophodne svim korisnicima IBM PC i kompatibilnih računara. Detaljan opis komandi sa svim opcijama koje su na raspolaganju sa pojedine verzije MS DOS a. Objašnjenje o primjeni. Posebne napomene za verziju 4.0.

125 strana — 230,00 din

2. PC Tools Deluxe 6.0

Priručnik za rad sa najpopularnijim pomoćnim programom za personalna računara. Knjiga sadrži opis instaliranja programa, korisničkog interfejsa, rada sa datotekama i direktorijima kao i objašnjenje za rad sa programima koje sadrži paket PC Tools. PC Shell, PC Backup, Compress, DiskFix, PC Cache, Desktop.

130 strana — 230,00 din

3. WordPerfect 5.1

Detaljan podsjetnik za rad sa tekst procesorom koji je osvojio svijet. Sve opcije programa su date abecednim redom što omogućava njihovo lako nalaženje. Opis rada korak po korak. Nove mogućnosti WordPerfecta: tabele i formule, stilovi i makro naredbe.

130 strana — 230,00 din

4. MS Quick Basic 4.5

Referentni priručnik za popularni programski jezik. Opis svih instrukcija i funkcija sa tačnom sintaksom i načinom pozivanja. Ilustrovano mnogobrojnim primjerima.

140 strana — 230.00 din

5. Norton Utilities

Priručnik za rad sa svim komponentama sistema Norton Utilities, od instaliranja programa i testiranja vašeg računara do korištenja naprednih mogućnosti ovog vrlo korisnog programa.

120 strana — 230.00 din

6. Lotus 123 Rel. 2.2/3.0

Podsjetnik za rad sa integrisanim paketom programa koji je nametnuo svjetski standard. Kompletan pregled svih komandi, opcija, funkcija i makroa programa.

150 strana — 230,00 din

7. Biblioteka Runtime funkcije za MS C

Knjiga koja će vam olakšati rad i uštedjeti trud i vrijeme za razvoj najčešće korištenih funkcija. Jednostavne naspram svakom C programeru. Detaljan opis više od 300 funkcija.

250 strana — 330,00 din

8. MS Windows 3.0

Windows serija!

Operativni sistem nove generacije. Budite među prvim koji će ovladati njegovim ogromnim mogućnostima.

160 strana — 250,00 din

9. Clipper 5.0

Pregled svih instrukcija, komandi i funkcija sa sintaksom, namjenom, opisom argumenata i primjerom korištenja. Podešavanje okruženja programa, instrukcije pretprocesora, standardne klase, funkcije extend sistema.

320 strana — 390.00 din

10. AutoCad 10

Referentni priručnik za rad sa popularnim programom namijenjenim dizajniranju i projektovanju uz pomoć računara u kom se na sistematičan način nude objašnjenja o radu sa svim komandama programa i raspoloživim opcijama.

280 strana — 390,00 din

11. Venture Publisher 3.0

Referentni priručnik za najpopularniji program iz oblasti stonog izdavaštva (DTP) na PC u. Instaliranje programa, sistematizovan pregled svih opcija, opis rada sa tabelama i formulama, instalacije fontova. Obuhvaćene su verzije za GEM i MS Windows.

250 strana — 390.00 din

12. CorelDRAW! 2.0

Windows serija!

Upoznajte se sa fascinantnim karakteristikama najboljeg programa za crtanje i dizajniranje na PC računarima. U novoj verziji donosi rad u 3D perspektivi, novijeg ce nove fontove, mogucnost kreiranja sopstvenih fontova, konverziju fontove u i iz raznih formata, nove trase programi i još puno, puno toga.

140 strana — 230,00 din

U pretplati (izlazi iz štampe 25.06.1991.)

13. QuattroPro 2.0

Referentni priručnik za rad sa jednim od najboljih programa za tabelarne proračune. Dat je pregled komandi, funkcija i makroa i opisan rad sa grafikom. Ako već radite sa programom QuattroPro ova knjiga će vam biti nezamjenjiv podsjetnik.

140 strana — 230,00 din

14. Word for Windows 1.1

Windows serija!

Upoznajte tekst procesor nove generacije. Iskoristite prednosti grafičke radne okoline WYSIWYG prikaza i štampanja, dinamičke izmjene podešaka sa ostalim Windows aplikacijama, moćnih makroa i ostalih pogodnosti ovog izvanrednog programa.

140 strana — 230,00 din

U pretplati (izlazi iz štampe 15.07.1991.)

15. MS Excel 3.0

Windows serija!

Ova knjiga će vas uvesti u sve tajne Excela, trenutno najmoćnijeg programa za tabelarne proračune. Jedini pravi izbor za korisnike koji traže snagu, ali i fleksibilnost i lakoću upotrebe. Pored osnova rada, komandi, funkcija i makroa programa, dat je i prikaz poređenih alata Solver i Q+E.

150 strana — 250.00 din

Knjige možete naručiti poštom - priloženom narudžbenicom ili telefonom na brojeve
071/270 213 i 071/621-025 od 09-17 sati

Computer hit biblioteka, Poštenski fah 116, 71210 Ilidža

N A R U D Ž B E N I C A

Naručujem sljedeće naslove iz edicije "Džepna knjiga za PC"

(upišite broj komada ispod odgovarajućeg broja knjige)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Prezime i ime _____

Adresa _____

Mjesto _____ Telefon _____ Fax _____

Troškove poštarine i pakovanja padaju na teret naručioca

INSTITUT ZA NUKLEARNE NAUKE "BORIS KIDRIČ", VINČA
CENTAR ZA PERMANENTNO OBRAZOVANJE
11000 Beograd, Nemanjina 4/X
Telefoni: (011) 683-390, 682-486, 641-155/107,181
Telefax: (011)682-486

KNJIGE U OBLASTI PROGRAMIRANJA, PROGRAMSKIH JEZIKA I PRIMENE RAČUNARA

1. AutoCAD (verzija 10.0)
konstruisanje i projektovanje pomoću personalnih računara
(šesto izdanje, 1991.)
Autori: Boris Damjanović i Petar Damjanović
Latinica, 444 strane, format B-5, cena 780 dinara

2. Uvod u C jezik
(četvrto izdanje, 1991.)
Autor: Vladan Vujičić
Latinica, 317 strana, format B-5, cena 585 dinara

3. Primena programa SYMPHONY na personalnim računarima
(treće izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 226 strana, format B-5, cena 455 dinara

4. OS/2 - vodič za korisnike
(prvo izdanje, 1989.)
Autor: Zorica Jelić
Latinica, 253 strane, format B-5, cena 455 dinara

5. VENTURA - računarsko izdavaštvo
(treće izdanje 1991.)
Autor: Predrag Davidović
Latinica, 253 strane, format B-5, izlazi iz štampe u novembru '91

6. FORTRAN 77
standard sa dopunama za personalne računare
(drugo izdanje, 1990.)
Autori: Vlajko Kocić i Zoran Konstantinović
Latinica, 422 strane, format B-5, cena 780 dinara

7. UNIX - vodič za korisnike
(drugo izdanje, 1990.)
Autor: Zorica Jelić
Latinica, 422 strane, format B-5, cena 780 dinara

8. Primena programa FRAMEWORK III na personalnim računarima
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 326 strana, format B-5, cena 585 dinara

9. PROGRAMSKI ALATI U MATEMATICI
MathCAD, Grapher, Eureka
(prvo izdanje, 1990.)
Autor: Ante Ćurlin
Latinica, 402 strane, format B-5, cena 715 dinara

10. Primena programa QUATTRO na personalnim računarima
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 296 strana, format B-5, cena 585 dinara

11. DOS ukratko
(prvo izdanje, 1990.)
Autor: Dragan Pantić
Latinica, 89 strana, format B-5, cena 260 dinara

12. Vodič za VAX/VMS
(prvo izdanje, 1990.)
Autori: Tamaš Kerepeš, Zvonko Orlolić, Saša Matijević
Latinica, 512 strana, format B-5, cena 910 dinara

13. Primena programa EXCEL na personalnim računarima
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 272 strane, format B-5, cena 585 dinara

14. UNIX - vodič za programere
(prvo izdanje, 1991.)
Autor: Zorica Jelić
Latinica, 326 strana, format B-5, cena 715 dinara

15. WINDOWS 3.0
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 273 strane, format B-5, cena 585 dinara

16. PRIMAVERA - upravljanje projektima uz pomoć računara
(prvo izdanje 1991.)
Autori: Jaroslav Urošević i Jelica Drašković Ostojić
Latinica, 365 strana, format B-5, cena 780 dinara

17. dBASE III+ priručnik
(prvo izdanje, 1991.)
Autor: Milorad Filipović
Latinica, 249 strana, format B-5, cena 559 dinara

18. Osnovi informaciologije i informacione tehnologije
(prvo izdanje, 1991.)
Autor: Ljubomir Dulović
Latinica, 538 strana, format B-5, cena 1001 dinar

19. LOTUS 1-2-3 (verzije 3.0 i 3.1)
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 411 strana, format B-5, cena 884 dinara

20. dBASE IV priručnik
(prvo izdanje, 1991.)
Autor: Ljubomir Lazić
Latinica, 306 strana, format B-5, cena 689 dinara

21. WORDPERFECT (verzija 5.1)
(prvo izdanje, 1991.)
Autor: Dragan Pantić
Latinica, 300 strana, format B-5, cena 689 dinara

22. Programiranje u CLIPPER-u 5.01
Autor: Alempije Veljović
Prvo izdanje, izlazi iz štampe u oktobru '91

23. FoxPro
Autor: Dušan Čašić
Prvo izdanje, izlazi iz štampe u oktobru '91.

"RAČUNARI", oktobar 1991.

Poručujem (ispod rednog broja knjige upisati broj poručenih primeraka)

Redni broj knjige	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.	23.
Broj poručenih primeraka																							

IME i PREZIME
(Naziv preduzeća) _____

ULICA I BROJ _____

BROJ POŠTE I MESTO _____ **TELEFON** _____

Porudžbenicu poslati sa petim primerkom uplatnice na adresu:
Institut za nuklearne nauke "Boris Kidrič", Vinča
Centar za permanentno obrazovanje, Beograd, Nemanjina 4/X.
Uplata se vrši na žiro račun:
INSTITUT ZA HEMIJSKU DINAMIKU I PERMANENTNO OBRAZOVANJE, broj: 60803-603-17361.
Troškove poreza na promet i isporuke knjiga snosi poručilac i namiruje ih unapred zajedno sa uplatom za knjige

U Novom Sadu 13—16 novembra
SAJAM RAČUNARSTVA
I INFORMATIČKIH KOMUNIKACIJA

Tradicionalna jesenja manifestacija u reprezentativnom izložbenom prostoru Sportsko poslovnog centra „Vojvodina" ove godine ima ambiciju da predstavi sveobuhvatnu tržišnu ponudu računarske tehnike i pratece opreme domace i strane programske podrške, informatičkih komunikacija, te birotehničkih ambijenata. Pošto se planira da više nego ikada ranije bude naglašen poslovni, privredni i prodajni karakter sajma zainteresovane firme upucujemo da se za dodatne informacije jave na tel 021/624-117 i 360-971 i fax 021/360-971

Specijalna ponuda!

Obaveštavamo vlasnike i korisnike PC racunara da preko „Jugoslovenske knjige" mogu izvršiti pretplatu na najkvalitetnije časopise iz ove oblasti po sledecim cenama za 1992 godinu

- **BYTE** (12 brojeva) po ceni od din **3.990,00**
- **PC MAGAZINE** (24 broja) po ceni od din **9.980,00**
- **ELEKTOR ELECTRONICS** (12 brojeva) po ceni od din **7.200,00**
- **PERSONAL COMPUTER WORLD** (12 brojeva) po ceni od din **8.400,00**
- **AMIGAWORLD** (12 brojeva) po ceni od din **4.140,00**

Cene su garantovane za pretplatnike koji uplatu izvrse do 30 oktobra 1991 godine
Uplata se može izvršiti opštom uplatnicom SDK na neš žiro račun broj 60804-601-12505 sa potpunim podacima

Molimo da uz naziv časopisa upišete napomenu („na adresu") ili („lično") u zavisnosti od toga da li zelite da vam časopis stiže poštom direktno od izdavaca ili da ga preuzimate u Knjizari „Jugoslovenska knjiga"

Radi uspešnijeg realizovanja porudzbina molimo da nam pošaljete kopiju uplatnice na adresu

„Jugoslovenska knjiga"
11000 Beograd
Knez Mihailova 2
(Kataloska prodaja)
Tel 621-567 (011)

gostol GOAP

servis

Naši ovlašćeni servisi su u Subotici, Vinkovcima, Slavonskom Brodu, Sisku, Bjelovaru, Rijeci, Varaždinu, Zaboku, Zagrebu, Karlovcu, Splitu, Zadru, Rijeci, Bujama, Slovenskoj Bistrici, Šmarje Sapu, Ljubljani, Ajdovščini i Novoj Gorici

proizvodi i usluge

IZ NAŠEG PROGRAMA NUDIMO VAM PO POGODNIM CENAMA KOMPLETNE RAČUNARE

- GOAP PC – AT 286-12/H20-65M 34.300 din
- GOAP PC – AT 286-12W/H40-20M 38.300 din
- GOAP PC – AT 286-16W/H40-70M 39.800 din
- GOAP PC – AT 386-16S/H40-28M 52.000 din
- GOAP PC – AT 386-25/H40-19M 77.500 din
- GOAP PC – AT 386-25C64/H40-19M 83.400 din
- GOAP PC – AT 386-33C64/H40-19M 103.900 din
- GOAP PC – AT 486-33/H40-19M 189.700 din

Za plaćanje u devizama dajemo dodatni popust
Cene su fco kupac



...postoje problemi koje može da reši samo

gostol GOAP

GOAP-GOSTOL, AUTOMATIZACIJA PROCESA, PRVOMAJSKA 124, 65000 NOVA GORICA, TEL.: (065) 26-511, 26-522, 23-411, FAKS: (065) 26-566

PRVI PUT NA NAŠEM TRŽIŠTU!

SAMPO

Generalni zastupnik za Jugoslaviju

DP JUGOTRADE NOVI SAD

Trikovicev trg br. 4
Tel. 021/27-234, 22-643, 622-033
Fax. 021/22-643

Poslovna banka
Societe Generale — Paris, France



IPS 386/25



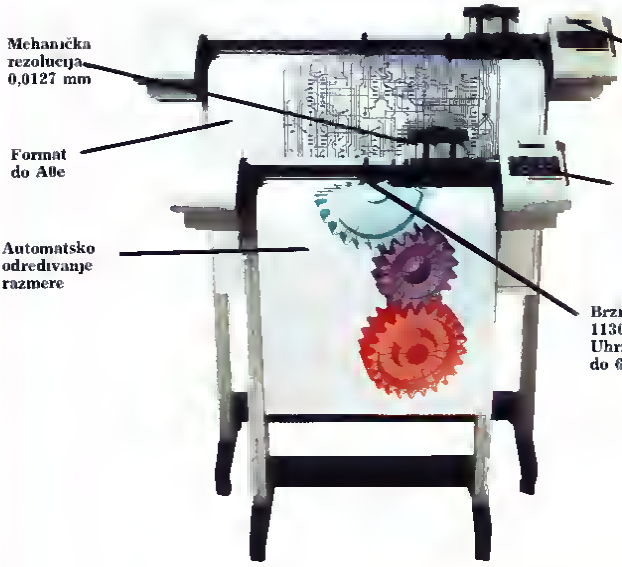


NOTE BOOK 386/SX

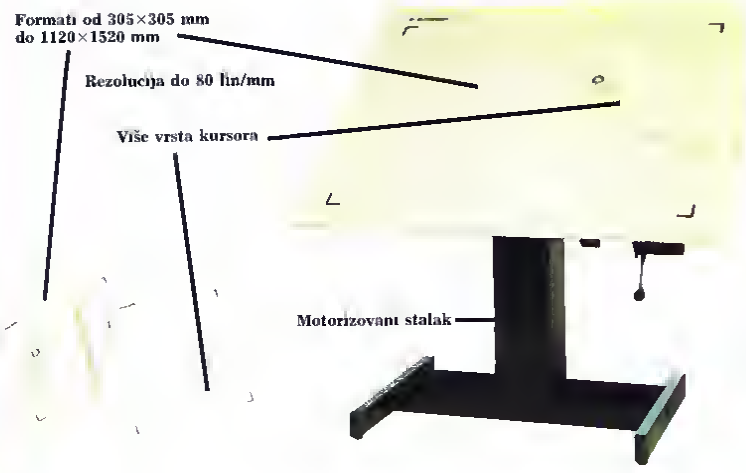
RAČUNARI I OPREMA	CENA U FRF (CIF BEOGRAD)
MICRO ORDINATEURS	
IPS 1515	3675.00
IPS 1535 — DD 20 Mo	5685.00
IPS 3515	4838.00
IPS 3535 — DD 20 Mo	6263.00
IPS 4615/5615	5250.00
IPS 4635/5635 — DD 40 Mo	8625.00
IPS 7715	6450.00
IPS 7735 — DD 40 Mo	9375.00
IPS 386 SX	7650.00
IPS 386 SX — DD 40 Mo	10425.00
IPS 386/25	16988.00
IPS 386/25—40	20400.00
IPS 386/25—70 — DD 70 Mo	22725.00
IPS 386/25—150 — DD 150 Mo	24675.00
PORTABLES	
PORTABLE LTL 286/16	16988.00
PORTABLE LTL 386 SX	17850.00
NOTE BOOK 386 SX	19575.00
MONITEURS MONOCHROMES	
BDS 1402 14" — 720×350 — Herc	893.00
BDS 1273 12" — 640×480 — VGA	818.00
BDS 1473 14" — 640×480 — VGA	1073.00
BDS 1902 19" — 1280×1024	4050.00
MONITEURS COULEURS	
KDS 1303 13" — 640×350 — EGA	2588.00
KDS 1453 14" — 640×480 — VGA	2618.00
KDS 1476 14" — 800×600 — Mult	3413.00
KDS 1484 14" — 1024×768 — VGA	2963.00
KDS 1488 14" — 800×600 — VGA	2813.00
KDS 1964 19" — 1024×768 — VGA	7800.00
KDS 2065 20" — 1280×1024 — HR	11700.00
KDM 2066 20" — 1280×1024 — Mult	13838.00
CARTES	
MGP mono Hercules	218.00
MIRA HR mono 1280×1024	3525.00
MIRAVGA couleur VGA	788.00
VGA 1024 couleur 1024×768	1073.00

HOUSTON INSTRUMENT
Summagraphics™

Ploteri



Digitalizatori



Skeneri

Sekači

Vrhunske grafičke periferije, softver i inženjering

Grafička tabla „Summagraphics“ na poklon! SVI KUPCI PLOTERA A0 i A1, SKENERA I SEKAČA DOBIJAJU GRAFIČKU TABLU „SUMMASKETCH II“ GRATIS!

Ovlašćene prodajne organizacije:

kma D.O.O.

Milentija Popovica 9
BEOGRAD (Sava centar)
Tel.: 011/222-19-28
Tel: 011/222-43-22, lok 615
Fax. 011/455-785

SoftWell d.o.o.
Kompjuterski inženjering
Pozorišni trg 1
21000 Novi Sad
Tel 021/51 999 52-996

El „Računari“ — Niš
Bul. V. Vlahovića 80—84
18000 Niš
Tel.: 016/325 461, 55 583
Fax. 325-726

Lanaco
Braća Pavlića H
78000 Banjaluka
Tel 078/33-992 46-717
Fax 33-992

Makpetrol kompjuteri
Kompjuterski inženjering
Ul. „Orce Nikolov“ bb
91000 Skopje
Tel 091/229-676, 221-040
Fax 229 676

EPSON

VELIKO IME, POUZDAN KVALITET!



COMPUTERS

EPSON-ove proizvode prodaju mode ovolim i

INTERCOP Beograd
ASVS COMMERCE Novi Sad
MIKRONOVA Novi Sad
INFOTRADE Petrinja

ELNOS Novi Sad, PARTNER Kruševac, DIVA Valjevo, Soccia, OMNIA Skopje, MAKPETROL Skopje, SEMOS Skopje



COMTRAD

VOLKS-AT 286/12

CONFIGURATION

- 80286 INTEL CPU & TI CHIPSET
- 12 MHZ SPEED
- LANDMARK 15.6
- 1 MB RAM INCLUDED
- 2 SERIAL PORTS
- 1 PARALLEL PORT
- 1 GAME I/O PORT
- SOCKET FOR 80287 COPOCESSOR
- FLOPPY DISC DRIVE 5.25" 1.2 MB
- FLOPPY DISC DRIVE 3.5 " 1.44 MB
- 40 MBYTE AT-BUS HARDDISC
- VGA DISPLAY ADAPTER
- 14" VGA COLOR MONITOR
- MF II KEYBOARD, 102 KEYS
- SERIAL MOUSE 2 BUTTOM

1.991.- DM

ISPORUKA ODMAH!



Build for your satisfaction

386SX (ista konfiguracija) 2.499 DM

SEIKOSHA SP 2000 Ai	SEIKOSHA SL 92A
9 iglica/192 cps	24 iglice/240 cps
DM 361	DM 595

COUPON

Please send me information about:

- 386 SX
- 386 CACHE 33
- 486 ISA
- 486 EISA 33

ComTrad
HIGH SPEED PC'S

ComTrad

COMTRAD WORLDWIDE SUPPORT
UNITED STATES • GERMANY • VENEZUELA
YUGOSLAVIA • NETHERLANDS • SPAIN

Central European Office
Wahlerstrase 12, Wahler Park
4000 Düsseldorf 30
West Germany
Tel: 99-49-211-65-08-0
Fax: 99-49-211-65-08-40

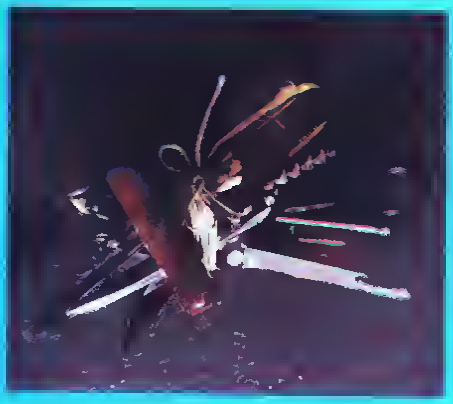
ComTrad Information
Vladimira Popovica 6
11000 Beograd
Tel: 011-2224-151
Tel: 011-2222-652
Tel: 011-2224-433/8265
Fax: 011-2223-951

ComTrad Zagreb
Krezmina 4
41000 Zagreb
Tel: 041-211-873
Fax: 041-211-873

KMA d.o.o.
Milentija Popovića 9
Tel: 011-2221 928
Tel: 011-2224-322,
lok. 615
Fax: 011-455-785


COMPA
works better.
ITORIZOVANI DISTRIBUTER I S

SVE što Vam je potrebno
u samo JEDNOM paketu





INFORMIX-4GL	**programski jezik četvrte generacije**
INFORMIX-SQL	**industrijski standard za upravljanje bazom podataka**
INFORMIX-4GL RDS	**interpreter četvrte generacije**
INFORMIX-ON LINE	**baza podataka za obradu velikog broja transakcija**
INFORMIX-NET	**omogućava distribuiranu obradu u mreži**

INFORMIX

GENERALNI
DISTRIBUTER ZA JUGOSLAVIJU

11070 Beograd, Lenjinov bulevar 12, P. Fah 20
Tel (011) 138-520, 139-712, 222-3072
• Fax (011) 222-4780 • TLX ENDATA YU 11764

Jednostavno,
radi bolje!



COMPAQ

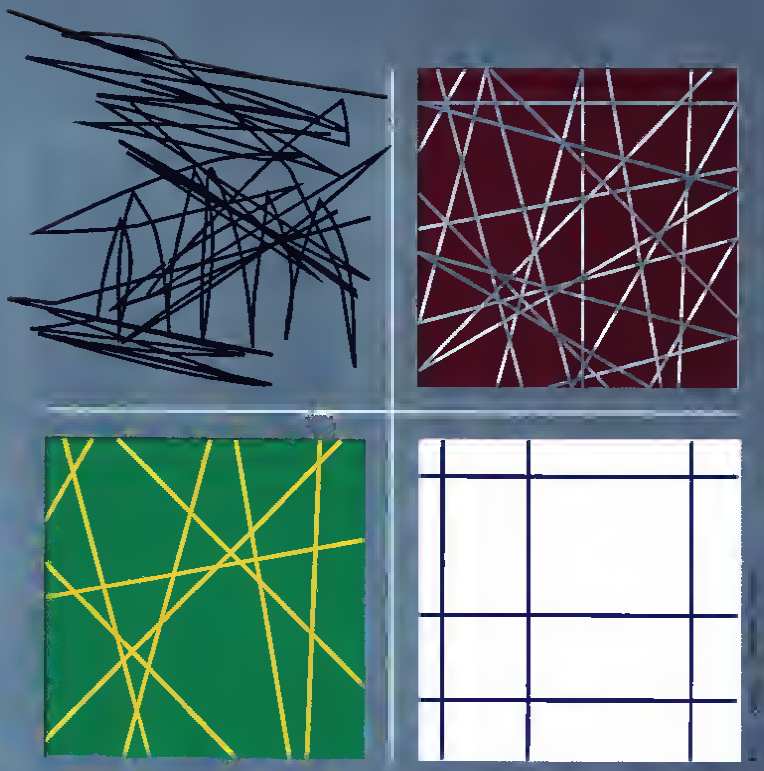
It simply works better.

AUTORIZOVANI DISTRIBUTER I SERVIS

MS MICROSYS

d.o.o. NOVI SAD Tel: 021/611-366 Fax: 021/20-544

POSLOVNICA BEOGRAD
DOBRAČINA 39 TEL/FAX (011) 622-581



The Art of Networking

Računare trebe povezati u mreže da bi se optimalno iskoristili ljudski i mašinski resursi. To je novi korak u razvoju računarstva i informatike. Vreme individualnih i velikih mainframe računara u poslovnom okruženju je prošlo.

Da bi se formirala mreže nije dovoljno samo kupiti kutije. Dođite kod nas sa znanjem o zahtevima koje sistem treba da ispuni u praksi i izradićemo projekat kompletnog informacionog sistema prema Vašim potrebama. Naši stručnjaci su kompetentni konsultanti koji mogu rešiti Vaše probleme u oblasti poslovne obrade. Poslovni softver koji vam nudimo predstavlja vrhunska dostignuća u ovoj oblasti.

Odabrali smo „Novell" najvećeg proizvođača softvera za lokalne računarske mreže, jer se „Most" pouzdaje samo u proizvođače koji garantuju kvalitet. Sa „Novell" NetWare operativnim sistemima naše mreže ispunjavaju sve Vaše zahteve za povezivanje PC mini i mainframe računara. Raspolažemo znanjem i iskustvom koje Vam garantuje da će Vaša mreža raditi uz maksimalno iskorišćenje performansi. Pored toga kod nas ćete dobiti kompletnu, stalnu podršku, otklanjanje svih kvarova u radu mreže u roku od 24 časa, kao i telefonsku podršku.

Kupovinom u „Mostu" tako dobijate kompletno rešenje Vašeg problema. Naši sistemi su projektovani tako da budu tehnički fleksibilni i finansijski povoljni. Obratite nam se zahtevom za ponudu za Vaš konkretan problem, ili zahtevajte listu naših referenci.



Zašto bi vaš novi kompjuter trebalo da bude "M O S T"

Kod nas se računari ne prodaju po svaku cenu – prodajemo samo najbolje. Ovde ćete dobiti savet, pomoć u radu ili projekat za kompletan informacioni sistem. Kompjuteri i informatika su naš životni poziv, pa možete biti sigurni da ćete kod nas uvek dobiti kompetentnu stručnu pomoć.

Ciljevi kompanije „Most" uvek su bili jasni: stvoriti bazu zadovoljnih korisnika, dostići i održavati visok nivo kvaliteta, postati najbolji. Mislimo da su ti ciljevi ostvareni, a vi ste tome doprineli. Danas, biti najbolji određuje koje će kompanije preživeti. „Most" će još dugo biti sa vama.

Nazovite telefonom – možda tako rešimo vaš problem.

Naš oglas više nije običan cenovnik. Mi se razlikujemo od drugih. Trudimo da sjedinimo umetnost i tehnologiju.

JUNIOR: 286/16 • ROSE: 386SX/20 • ASSISTANT: 386/33 • GENIUS: 386/40
DOLPHIN: 486SX/20 • STAR: 486/25 • GOLIATH: 486/33

NOVO: NetWare-Ready diskovi – tvrdi diskovi spremni za instalaciju »Novell« mreža, se obavljenom analizom površine diska.

MOST

WordPerfect® familija

WordPerfect

vodeći tekst procesor na svjetskom tržištu
preko 4,000 000 korisnika
jednostvena sprega snage i jednostavnosti
preko 500 podržanih printera
tabele, matematika, jednadžbe
prethodni pregled dokumenta
integracija grafike u tekst

DataPerfect

relacijska baza podataka
jednostavno generiranje ekrana
jednostavno generiranje izvještaja
fleksibilni pregled podataka
run-time modul

WordPerfect OFFICE

integracija svih WP produkata
potpuno rješenje automatizacije ureda
pristup do drugih aplikacija
elektronička pošta (e-mail)
grupni terminski kalendar
WAN i globalna komunikacija
kalkulator, rokovnik



domaća verzija!!!

PlanPerfect

moćan tabelarni kalkulator
izvanredne grafičke mogućnosti
optimizacija preračunavanja
preko 2,000,000 ćelija po tabeli
preko 100 specijaliziranih funkcija
upotreba virtuelne memorije
povezivanje tabela

DrawPerfect

vrhunski paket za poslovnu grafiku
preko 500 predefiniranih slika
tabele, grafikoni
vektorska definicija objekata
ulaz podataka iz 14 grafičkih formata
izlaz na jednake visoke rezolucije
30 različitih fontova
run-time modul

SAMO NAJBOLJE JE DOVOLJNO DOBRO!

perpetuum d.o.o.
integralni informacijski sistemi
Kučerina 5, 41000 Zagreb tel 317-020 328-999, fax.328-999

Ploteri Océ Graphics
Evropa je odabrala kvalitet



Izbor naših plotera je veoma širok! (na žalost ova stranica ni

Ploteri sa perima svih formata
Termalni ploteri
Grafičke table i digitajzeri
Color POSTSCRIPT termalni printeri

Najkonkurentnije cene vodećeg evropskog proizvođača
Širok izbor potrošnog materijala
Obezbeden servis



océ Graphics

Ekskluzivni distributer:

Računarski inženjering

Beograd, tel: (011) 557-592; fax. (011) 559-845

OVLAŠĆENE PRODAJNE ORGANIZACIJE

„FONI"
11300 Smederevo
Tel/fax (026)24-544

„BEE LINE"
18000 Niš
Tel (018)62 063
Fax (018)47-424

„COBEX"
81000 Titograd
Tel/fax (081)24 148

„ARDI ENGINEERING"
38000 Priština
Tel (038)22 581
Fax (38)23-578



mikronova

DELATNOST - Projektovanje i izgradnja informacionih sistema

- PRODAJA NOVELOVIH PROIZVODA
- PROJEKTOVANJE informacionih sistema
- PROIZVODNJA kompletnog hardware-a (računarske opreme)
- PROJEKTOVANJE I IZGRADNJA računarskih mreža i komunikacija
- IZRADA software-a (aplikativnih programa)
- ODRŽAVANJE I SERVISIRANJE harware-a i software-a
- NABAVKA repromaterijala (diskete, riboni, perforirani papir ..)
- OBUKA korisnika

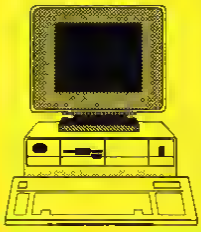
NOVELL® AUTHORISED DEALER

NOVELL®

RAČUNARSKE MREŽE

- NETWARE Ver 2.2
- NETWARE Ver 3.11
- MREŽNE KARTICE I OSTALA MREŽNA OPREMA

HARDWARE



Proizvodnja i isporuka najsavremenije računarske opreme, u konfiguracijama prema zahtevu korisnika:

- PC/AT-286 kompatibilni računari
- PC - 386 kompatibilni računari
- PC - 486 kompatibilni računari

- Hard diskovi
- Flopi diskovi
- Grafičke karte
- Monitori visoke rezolucije
- FAX karte i Modemi
- Matrični, laserski i heavy duty štampači

SOFTWARE

Proizvodnja optimalno koncipovanog aplikativnog software-a za IBM PC/XT/AT - kompatibilne računare.

- OPTIMALNA UPOTREBA POLJOPRIVREDNIH MAŠINA
- MATERIJALNO I FINANSIJSKO POSLOVANJE
- KADROVSKA EVIDENCIJA, LIČNI DOHOTCI I KREDITI
- UPRAVE DRUŠTVENIH PRIHODA
- KLINIKE I BOLNIČKA POSLOVANJA,
- GINEKOLOŠKO AKUŠERSKE KLINIKE
- STAMBENA PREDUZEĆA
- UNIVERZALNI PAKETI ZA STATISTIKU, TRGOVINSKA PREDUZEĆA, ADVOKATSKE KANCELARIJE, OBRAČUN KAMATA, BUDŽETSKO KNJIGOVODSTVO
- UNIVERZALNI GENERATOR IZVEŠTAJA
- PROGRAMI PO ZAHTEVU KORISNIKA

NOVO

CLIPPER USER KLUB U OSNIVANJU!
Sva najnovija dostignuća i informacije direktno od NANTUCKET-a, javite se!

PROVERITE NAŠU REFERENTNU LISTU

DILERI DOBRODOŠLI !



NAŠA NOVA ADRESA JE:
P.P.mikronova D.O.O. Novi Sad, Lenjinov trg 5/II
Tel.021/52-999 Fax.021/28-042

MACE RAČUNARI P.O.

Zmaj Jovina 4/II/1, 11000 Beograd
Tel/Fax (011) 189-455

NAJNIŽE SVETSKE CENE PC RAČUNARA

IZVOD IZ CENOVNIKA - OKTOBAR 1991.

386SX-16 C&T	400
386SX-20 C&T	497
386-25 OPTI	759
386-33 64K cache SIS	994
486SX-20 64K cache PEAK-DM	1394
486-25 64K cache PEAK-DM	1725
486-33 64K cache PEAK-DM	1863
SVGA 16/512K (1024x768) OAK067	121
SVGA 16/1M (1024x768) ET4000EX	221
14" TVM SVGA 1024x768 MONO	359
14" A— SVGA 1024x768 COLOR	682
PRINTER STAR 9PIN, A4	522
PRINTER HYUNDAI 9PIN, A3	551
PRINTER CITIZEN 24PIN, A4	575
PRINTER PANASONIC 24PIN, A4	756
PRINTER EPSON 24PIN, A4	814
PRINTER NEC 24PIN, A3	1239
HP LASERJET IIIP	2732
PLOTER SECONIC A3	1515
MODEM 1200	137
MODEM 2400	174
MODEM 2400 E	211
MODEM 2400 I MNP5	248
MODEM 2400 E MNP5	273
SKENER GENIUS C/B 400TPI	262
SKENER GENIUS COLOR 400TPI	825
287-12 AMD	275
287-20 IIT	549
387SX-16 IIT	564
387SX-20 IIT	620
MOUSE GENIUS GM6	30
MOUSE SMII CORDLESS	190
TRECKBALL GENIUS GK-T320	163

MBOARD	CLOCK	RAM	FD	BEZ HD	43M/25	86M/19	120M/19	212M/19
286	12MHz	1M	1	1020	1430	1810	2080	2730
286	16MHz	1M	1	1190	1520	1970	2250	2980
386SX	16MHz	2M	1	1270	1680	2060	2330	3060
386SX	20MHz	2M	1	1360	1780	2150	2420	3610
386DX	25MHz	4M	1	1820	2230	2610	2880	4010
386DXc	33MHz	4M	2	2220	2630	3010	3280	4410
486SXc	20MHz	4M	2	2620	3030	3410	3680	4740
486DXc	25MHz	4M	2	2950	3360	3740	4010	4880

U NAVEDENE CENE JE UKLJUČEN TTL MONITOR, US TASTATURA 102 TASTERA, MINI TOWER, IDE FD/HD KONTROLOR, PORTOVI

Uz svaki računar	BESPLATAN MIŠ
Uz svaki štampač	BESPLATAN KABL
Uz svaki računar sa štampačem	BESPLATAN MODEM
Uz svaku disketnu jedinicu	BESPLATNA KUTIJA DISKETA
Uz svaki SVGA kolor podsistem	BESPLATAN DŽOJSTIK

DOPLATE ZA

SVGA 1024x768 MONO PODSISTEM (MONITOR+KARTA)	240
SVGA 1024x768 KOLOR PODSISTEM (MONITOR+KARTA)	560
DISKETNU JEDINICU (1.2M 5.25" ILI 1.44M 3.5")	140
1M MEMORIJE	100
TOWER KUĆIŠTE	80

Sve cene su u DEM. Uračunate su sve propisne dažbine (carina, transport...) Besplatna dostava u Beogradu. Plaćanje 100% avansa, rok isporuke do 30 radnih dana. Garantni rok za ceo sistem nezavisno od veličine 18/24 meseci, za komponente 3-12 meseci. Vanjamino održavanje. Sklapanje računara od donetih komponenti ugradnja donetih komponenti instalacija donetog diska, paušalno 100 - DEM.

ELITE & EUROHIT Computers

BEOGRAD, Vodovodska 80 Tel./Fax. (011) 516-253

MI SMO SIGURNI DA SU KVALITET I DOBRA CENA USLOV ZA OPSTANAK!

386

STANDARDNA KONFIGURACIJA:

- **HD 42 MB**
- **1 MB RAM**
- **FLOPI DISK 1,2 MB**
- **TASTATURA KLIK,**
- **KUĆIŠTE, PS 200 W**
- **2P/2S**

PC AT 386/25 MHz, Mono — 2299 DEM

PC AT 386/25 MHz, VCA
Monitor VCA kolor 1024x768
VCA grafička karta, 16 bit-a, 512 KB — 3099 DEM

PC 386/33 MHz
64 KB CACHE, Mono — 2699 DEM

PC 386/33 MHz
64 KB CACHE, VGA — 3499 DEM

Panasonic

PRINTERI PANASONIK - 24 pina

A3 240 cps	1250 DEM	Co - procesor 387/25 MHZ	629 DEM
Modem 2400 bps	240 DEM	Co - procesor 387/33 MHZ	699 DEM
Fax-Modem karte	550 DEM	Miševi	80 DEM

Servis i garancija obezbeđeni

Doplata za

• 1 MB Ram	120 DEM	• HD 80 MB	400 DEM
• HD 128 MB	800 DM	• Flopy 3,5 " 1.44 MB	200 DEM

Kod nas prvo proverite robu, pa tek onda plaćate u dinarskoj protiv vrednosti
ISPORUKA ODMAH - SVE CENE SU ZA KRAJNJEG KUPCA

AUTODESK

AutoCAD R.11
AEC
AutoSHADE
AutoSKETCH
ANIMATOR
3D STUDIO

océ Graphics

PLOTERI SA PERIMA
TERMALNI PLOTERI
GRAFIČKE TABLE
DIGITAJZERI
KOLOR TERMALNI
PRINTERI

RASTEREX Graphics solutions

PROFESIONALNE
GRAFIČKE KARTICE:
LIGHT AT,LIBERTY
AutoIcon
RxImage
SpotLight

genoa

OSNOVNE PLOČE:
386-33 MHz/64k,
486-33 MHz/256k
GRAFIČKE KARTICE



Chicony

TASTATURE

LAPTOP RAČUNARI

PROJEKTOVANJE SISTEMA
PC RAČUNARI SVIH KONFIGURACIJA
NOVELL MREZE
AutoCAD I POSLOVNE APLIKACIJE

RAČUNARSKI INŽENJERING
BEOGRAD

RAČUNARSKI INŽENJERING
COMPUTER ENGINEERING

Sedište/Head office
Mice Stojkovica 8,
11130 Beograd

Predstavništvo/Representative office
Šumadijski trg 6a,
11030 Beograd,

tel. 38 11 557-592
fax. 38 11 559-845

MCH

AUTRONIC

MCH Computer-Systeme
Handelsgesellschaft m b H
8472 Strass/Stmk Hofgreith 2
Tel 9943 34 53 44 50
Fax 9943 34 53 43 65

PEACOCK COMPUTER

AUTRONIC Computer Systeme
A 9020 Klagenfurt Radetzkystr 18
Tel 9943 463 51 48 71
Fax 9943 463 51 48 73

Osnovna konfiguracija

AT 286 - 16	1 MB RAM, VGA , 40 MB tvrdi disk, DOS 4 01	47 821 DIN
AT 286 - 16	1 MB RAM, VGA , 110 MB tvrdi disk, DOS 4 01	63 279 DIN
AT 386 - 25	2 MB RAM VGA , 40 MB tvrdi disk DOS 4 01, WINDOWS 3 0 + MS Mouse	81 282 DIN
AT 386 - 25	2 MB RAM VGA, 110 MB tvrdi disk DOS 4 01, WINDOWS 3 0 + MS Mouse	96 740 DIN
AT 386 - 25C	4 MB RAM VGA, 110 MB tvrdi disk DOS 4 01, WINDOWS 3 0 + MS Mouse	114 153 DIN
AT 386 - 33C	4 MB RAM, VGA, 110 MB tvrdi disk DOS 4 01 WINDOWS 3 0 + MS Mouse	127 159 DIN
AT 486 - 25C	4 MB RAM, VGA, 110 MB tvrdi disk, DOS 4 01 WINDOWS 3 0 + MS Mouse	163 259 DIN
MONITOR	VGA MONO MONITOR 640 x480	5 941 - DIN
MONITOR	VGA TRISCAN MONITOR 1024 x 768	22 843 DIN
	NOTEBOOK 80386 SX	**136.188,- DINARA**

Distributeri

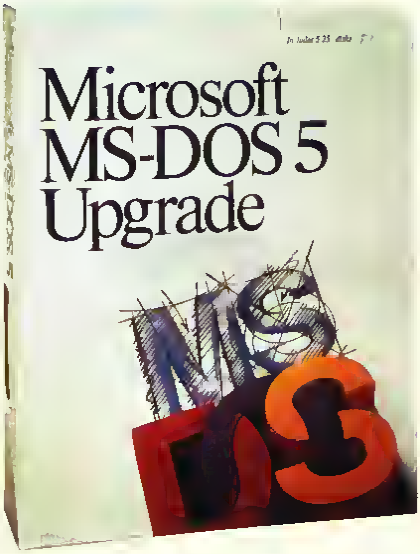
MCH Computer d.o.o.
62000 Maribor, Tomšičeva 19, Tel & Fax (062) 28 250
MCH Solution d.o.o.
11000 Beograd, Zaplanjska 86, Tel (011) 468 732
Fax (011) 467 059
MCH Tehnologies d.o.o.
41000 Zagreb, Proleterskih brigad 78, Tel (041) 539 892,
Fax (041) 538 946

AUTRONIC d.o.o.
61000 Ljubljana Kardeljeva ploščad 17
Tel (061) 345 161
Tel & Fax (061) 302 581
AUTRONIC d.o.o.
41000 Zagreb, Kollerova 3
Tel & Fax (041) 232 259



PRELAZNI ROK

Pređite još danas sa nepotpune i zastarele piratske kopije na novi Microsoft DOS 5.0 pomoću ...



Microsoft DOS 5.0 Upgrade je na raspolaganju svim korisnicima, koji već imaju MS-DOS na PC računaru. Takođe onima koji su dosada radili na piratskoj kopiji.

Potpuno nova verzija Microsoftovog operacionog sistema koji inače pokreće 60 miliona PC računara širom sveta sa sobom donosi:

YU slova – napokon su naša slova originalno podržana od strane operacionog sistema sa codepage 852
Task Switcher – prelaz iz jedne u drugu aplikaciju jednim pritiskom na tastaturu, više memorije (do 620 K) na raspolaganju za vaše aplikacije
DOS Shell – rad sa fajlovima preko jednostavnog i preglednog user interfejsa
QBasic – moderan i strukturiran Basic programski jezik
Editor – ekranski editor teksta

Prodaja Microsoft DOS 5 Upgrade i ostale programske opreme Microsoft sa registracijom u Jugoslaviji

CET Beograd, tel 011/336-630
Jugodata Beograd, tel 011/453-560
REY Beograd, tel 011/488-5472
Softwell Novi Sad, tel. 021/51-999
Mladinska knjiga Zagreb, tel 041/422-460
INTES zagreb, tel 416-162
MicroLab Zagreb, tel 041/692-703
Almido Split, tel 058/43-137
Print Zagreb,tel 041/412 843
Medija Ljubljana, 061/221-838
7 L Murska Sobota, tel 069/21-353
Mikrohit Ljubljana, tel 215-042
CAT Ljubljana, tel 061/223-949
SMN Nova Gorica, tel 065/26-800
Monitoring Zagorje, tel 0601/81-935

Za sve kupce do 15. oktobra nagradno izvlačenje Microsoftovih proizvoda na štandu Atlantis na sajmu INTERBIRO u Zagrebu

Distribucija Microsoft programske opreme u Jugoslaviji

ATLANTIS d.o.o.
POSLOVNI INFORMACIJSKI SISTEMI
Cankarjeva 4
61000 Ljubljana
tel/fax
(061) 221 608

ATLANTIS

Microsoft je zaštitni znak Microsoft, Inc